Marika Kästner

Optimierung im Workforcemanagement - mathematische Modellierung und Lösungsstrategien

Hochschule Mittweida

University of Applied Sciences

Fachbereich Mathematik

Mittweida, Juli 2010

Marika Kästner

Optimierung im Workforcemanagement - mathematische Modellierung und Lösungsstrategien

eingereicht als

MASTERARBEIT

an der

HOCHSCHULE MITTWEIDA

UNIVERSITY OF APPLIED SCIENCES

Fachbereich Mathematik

Mittweida, Juli 2010

Erstprüfer: Prof. Dr. R. Fischer
Zweitprüfer: Herr Dipl.-Ing. T. Epler

Die vorgelegte Arbeit wurde verteidigt am: 08.07.2010

Bibliographische Beschreibung:

Marika Kästner:
Optimierung im Workforcemanagement - mathematische Modellierung und Lösungsstrategien. - 2010. - 123 Seiten Hochschule Mittweida, Fachbereich Mathematik/Naturwissenschaft/Informatik, Masterarbeit, 2010

Referat:
Innerhalb dieser Masterarbeit wird die Automatisierung der Disposition bei der Firma envia NSG als Optimierungsproblem beschrieben. Im ersten Teil findet eine Problemspezifikation im Hinblick auf der Umsetzung mit Fuzzy-Optimierung statt. Dabei werden mehrere Optimierungsziele formuliert und einige Besonderheiten herausgestellt. Im zweiten Teil der Arbeit werden Modelle und Lösungsstrategien für dieses Problem vorgestellt, welche auf der diskreten Optimierung mit nur einem Optimierungsziel beruhen.

Inhaltsverzeichnis

Selbstständigkeitserklärung

Hiermit erkläre ich, dass die vorliegende Masterarbeit selbständig und nur unter Verwendung der angegebenen Hilfsmittel angefertigt worden ist.

Mittweida, den 25. Juli 2010

Marika Kästner

1 Einleitung

Die envia Netzservice GmbH, kurz envia NSG, ist eine hundertprozentige Tochterfirma der enviaM, dem größten Energieversorger in den neuen Bundesländern Deutschlands. Das Versorgungsgebiet von envia NSG erstreckt sich von West-Sachsen über Süd-Brandenburg, dem östlichen Gebiet in Sachsen-Anhalt und einem kleinen Teil Thüringens. Die Gesamtfläche beträgt ca. 25.912 km^2. Insgesamt stehen etwa 340 Einheiten im gesamten Raum zur Verfügung, die für das Beseitigen von Störungen, Erledigen von Kundenwünschen, Wartung und Instandhaltung in den Netzanlagen, Netzstationen und Umspannwerken verantwortlich sind.

Das Auftragsvolumen eines Jahres beträgt mehr als 150.000 [2]. Jeden Tag müssen die einzelnen Aufträge einer Einheit, bestehend aus einem Mitarbeiter, Fahrzeug und weiteren Ressourcen, unter Beachtung aller Nebenbedingungen zugeordnet werden. Für diese umfangreiche Aufgabe sind die Dispatcher verantwortlich.

Um die Dispatcher zu unterstützen und diverse Kosten zu minimieren, wird enviaM eine Optimierungssoftware einführen, welche eine automatische Auftragseinplanung und Vordisposition vornimmt. An die Optimierungssoftware werden viele Forderungen gestellt. So sollte zum Beispiel eine Routenoptimierung stattfinden, so dass Fahrzeit gespart und damit der Anteil an Arbeitszeit erhöht wird. Zudem sollte die Software in der Lage sein, immer unter Beachtung aller Nebenbedingungen, Lücken zwischen zwei Terminaufträgen mit Hilfe von weiteren Einsätzen optimal zu schließen.

Da, wie später noch deutlich wird, der Umfang des zu lösenden Optimierungsproblems sehr komplex ist, wurde die Firma *Fuzzy Logic Systems GmbH* (F/L/S) beauftragt, die in der Automatisierungstechnik eingesetzte Software für die Aufgabenstellung der envia NSG anzupassen. Im Rahmen dieser Masterarbeit werden die Ziele und Aufgaben der Anpassung erläutert. Da den Verfahren von F/L/S die Fuzzy-Optimierung zugrunde liegt, werden die Optimierungsziele und Nebenbedingungen im ersten Teil der Arbeit auch auf dieser Basis formuliert.

Im zweiten Teil der Arbeit wird das Ausgangsproblem mit den betrieblichen Forderung aus einem anderen Blickwinkel betrachtet und eine alternative Herangehensweise vorgestellt. Die nun verwendeten Methoden beruhen auf bekannten mathematischen Verfahren. Auf eine Umsetzung dieser Methoden wird jedoch verzichtet, da bereits eine Umsetzung mit Hilfe von *Qualicision®* von F/L/S geschieht.

Im letzten Kapitel werden der Inhalt der Masterarbeit zusammengefasst und die wichtigsten Resultate herausgestellt.

Die Masterarbeit unterteilt sich in die folgenden Kapitel:

Begriffe und Prozess der envia NSG Am Anfang wird eine Einführung in die Begriffe und Prozesse der envia NSG gegeben. Diese bilden die Grundlage für die Problemspezifikation.

Problemspezifikation Aus den allgemein beschriebenen, betrieblichen Zielen werden einzelne Optimierungsziele mit den dazugehörigen Nebenbedingungen formulierten.

Mathematische Grundlagen und Methoden der Optimierung Um die alternative Herangehensweise verständlich zu beschreiben, werden einige mathematischen Grundlagen benötigt. Diese werden in diesem Kapitel erläutert und zudem auf die Algorithmen später benötigter Optimierungsverfahren eingegangen.

Alternative Herangehensweise Nachdem das Problem für die, auf der Fuzzy-Optimierung basierenden, Software spezifiziert wurde, wird nun ein alternative Herangehensweise dargelegt. Dabei wird das Problem von einem anderen Blickwinkel betrachtet. Die so modellierten Probleme gehören der Diskrete Optimierung an und lassen sich mit Modifikation bekannter Methoden lösen. Die Lösungsmethoden für die einzelnen Probleme werden vorgestellt und näher erläutert.

Zusammenfassung und Ausblick Am Ende der Arbeit werden die wesentlichen Resultate der Masterarbeit noch einmal hervorgehoben und ein Ausblick für weiterführende Arbeiten angegeben.

2 Begriffe und Prozess der envia NSG

Die envia NSG ist verantwortlich für den Bau, die Wartung, den Betrieb, die Instandhaltung und die Betriebsführung der Verteilernetze der enviaM. Zu den Aufgaben zählen unter anderen Entstörungen, Wartungen, Inspektionen und Instandsetzungen.

Am 01.01.2005 trat innerhalb der envia NSG ein neues Geschäftsmodell in Kraft. Die Erneuerungen dienen vor allem der Kostensenkung und der damit verbundenen Effizienzsteigerung. Die Effizienzsteigerung wurde unter anderem durch einen optimierten Personaleinsatz, der Automatisierung von Prozessen und der Aufhebung der regionalen und organisatorischen Trennung nach Spannungsebenen erreicht. Im Januar 2008 wurde eine zentrale Dispatchingorganisation eingeführt [1].

Die tägliche Aufgabe des Dispatchers besteht darin, Aufträge zu den einzelnen Einheiten unter Berücksichtigung von Nebenbedingungen, wie zum Beispiel den Ressourcen, einzuplanen. Mit der Einführung des neuen Geschäftsmodells und der damit verbundenen Forderung der überregionalen und spannungsübergreifenden Steuerung des operativen Betriebes, wurde die Aufgabe des Dispatcher somit weitaus umfangreicher. Jedem Dispatcher ist ein größeres Gebiet mit mehr Außendienstmitarbeitern unterstellt.

Da eine solche komplexe Aufgabenstellung und die Verwaltung bzw. Pflege der Daten manuell kaum zu bewältigen sind, bedarf dies einer Unterstützung durch ein Workforcemanagement-System (WFM-System). Der Begriff Workforcemanagement wird in vielen verschiedenen Hinsichten in der Wirtschaften verwendet. Aus diesem Grund werden an dieser Stelle die Definitionen, welche auf das Unternehmens enviaM zutreffen, angegeben.

Definition 1 (Workforcemanagement (WFM))**.** *(seitens enviaM) [2]*
Workforcemanagement dient zur Unterstützung der planbaren und nicht planbaren Aktivitäten bei Baumaßnahmen, Inbetriebnahmen, Betrieb, Instandhaltung und Störungsbeseitigung in Versorgungsnetzen zur kostenoptimalen Ressourcensteuerung (Personal, Werkzeug und Material) und Ergebnisdokumentation.

Definition 2 (Workforcemanagement-System (WFM-System))**.** *(von PSE (Siemens)) [5]*
Workforcemanagement-Systeme sind intelligente Systeme, die permanent, also in Echtzeit, aktuelle Änderungen berücksichtigen können. Der Kontakt zwischen Mitarbeitern und dem System kann auf vielfältige Weise hergestellt werden, z. B. über Internet oder Telefon.

Innerhalb der envia NSG wurde das Ressourcen-Management-System (RessMa) eingeführt. Dieses dient der operativen Unterstützung bei der Umsetzung von Entstörungs- und Instandhaltungsmaßnahmen im Versorgungsnetz der envia NSG. RessMa wirkt als Auftrags- und Einsatzleitsystem und unterstützt somit die Prozesse in der operativen Netzbetriebsführung in allen Spannungsebenen (Hoch- (HS), Mittel- (MS) und Niederspannungen (NS)) [3]. RessMa ist dabei nur ein Bestandteil vieler weiterer des gesamten Workforcemanagement-Systems. Die Übersicht der Wirkungsweise der einzelnen Systeme ist in Abbildung 2.1 dargestellt. Die Pfeile geben dabei die

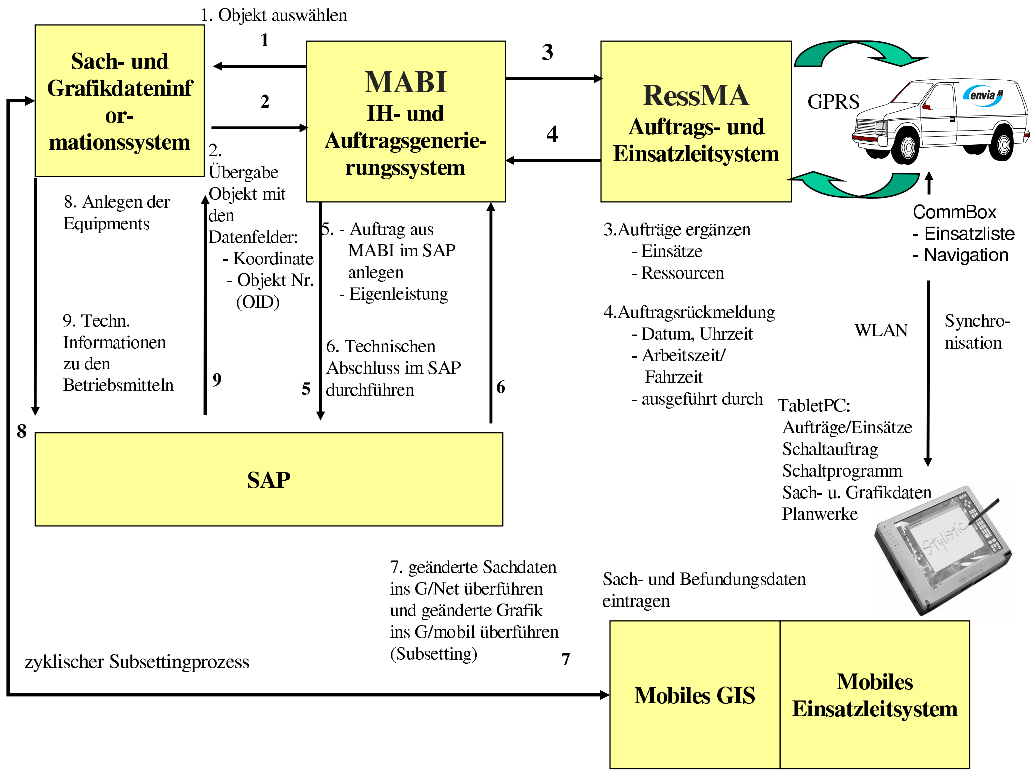


Abbildung 2.1: Das gesamte WFM-System der enviaM [3]

Kommunikation zwischen den einzelnen Systemen an. Die Konzentration wird auf das Auftrags- und Einsatzleitsystem (RessMa) mit den Schnittstellen zum Instandhaltungs- und Auftragsgenerierungssystem (MABI) und zur mobilen Einheit (dargestellt durch das envia NSG-Fahrzeug) gelegt.

RessMa unterstützt den Dispatcher bei seiner täglichen Arbeit. Im Folgenden wird die Umfang dieser Aufgaben verdeutlicht und ausgewählte Abläufe erläutert. Alle Begriffserläuterungen beziehen sich auf das Unternehmen enviaM und sind übersichtsmäßig nochmal auf den Seite 108ff. zusammengefasst.

Ein Auftrag wird im modularen Anlagen-Bewertungs- und Instandhaltungspaket (MABI) erzeugt und an RessMa übergeben. Das elektronische System MABI dient unter anderem zur Unterstützung der Auftragserzeugung und -abwicklung. Zu jedem Auftrag sind verschiedene Informationen gegeben: Priorität und damit verbundene Terminvorgaben, Status, erforderliche Ressourcen, Auftragstyp, zugehörige Vorgänge, Ausführungsort und viele weitere Informationen, welche für die Ausführung benötigt werden.

Es existieren die folgenden fünf Auftragsprioritäten mit der Rangfolge:

1 Störung

2 Terminauftrag

3 termingebundener Auftrag

4 Turnusauftrag

0 nicht priorisiert

Tritt eine Störung im Netzgebiet auf, so wird umgehend eine Einheit gesucht, welche frei ist bzw. ihre derzeitigen Arbeit unterbrechen kann und welche sich im nahen Umfeld des Netzobjektes mit der Störung befindet. Eine Störung hat oberste Priorität. Sie wird vom Dispatcher schnellst möglichst bearbeitet. Es gibt somit keine lange Planungsphase.

Bei Terminaufträgen ist ein Ausführungszeitpunkt festgelegt. Terminaufträge stehen in der Priorität direkt unter den Störungen und sind unbedingt einzuhalten. Der Unterschied zwischen Terminaufträgen und termingebundenen Aufträgen ist, dass termingebundene zunächst nicht einem fixen Datum zugeordnet sind. Oft werden aus termingebundenen Aufträge nach ihrer Einplanung Terminaufträge. Ein typisches Beispiel dafür sind Aufträge mit Schaltantrag (SAN). Der Dispatcher ordnet diesen an einen Tag ein und beantragt die Schaltung bei der Schaltleitungszentrale. Nach der Genehmigung der Schaltung wird aus dem termingebundenen ein Terminauftrag. Eine Schaltung kann mit Versorgungsunterbrechungen der zugehörigen Stromentnehmer der betreffenden Netzanlage verbunden sein und stellt aus diesem Grund eine Besonderheit dar.

Aufträge mit geringer Priorität sind Turnusaufträge und solche, die nicht priorisiert sind. Turnusaufträge sind meistens Inspektionen bzw. Befundungen und nur mit einem Ausführungszeitraum versehen. Unterschiedliche Arten an Turnusaufträgen haben unterschiedliche Turnusse. So müssen zum Beispiel Hausanschlüsse aller 6 Jahre inspiziert und Leistungsschalterwartungen aller 2 Jahre durchgeführt werden. Der Ausführungszeitraum gibt an, wann ein solcher Auftrag frühsten und bis wann spätestens bearbeitet werden sollte.

Jeder Auftrag ist wiederum in Vorgänge unterteilt. Das Augenmerk wird dabei auf den Hauptvorgang gelegt, da dieser die eigentliche Aufgabe enthält. Nebenvorgänge sind meistens Vor- oder Nachbearbeitungen oder Schalthandlungen. Diese besitzen dann eine Relativzeit zum Hauptvorgang, also eine Zeitspanne, wann diese vorher bzw. nachher erledigt werden müssen. Dazu ein kleines Beispiel: Ein Auftrag X lautet Wartung der MS-Anlage Y. Der eigentliche Hauptvorgang ist dabei die Wartung der Anlage. Da bei der Wartung einer Anlage zusätzlich diese abgeschaltet werden muss, wird vorher ein Schaltantrag gestellt (mindestens 7 Tage vorher), die Abstellinformationen an die betroffenen Kunden verteilt (2-3 Tage vorher) und die Schalthandlung muss vor dem Beginn der Wartung geschehen. Nach Beendigung des Hauptvorganges erfolgt die Rückschaltung. Falls Auffälligkeiten bei der Wartung festgestellt werden, werden diese wenn möglich vor Ort behoben. Somit ergeben sich insgesamt neben dem Hauptvorgang fünf Nebenvorgänge.

Der Aufgabenbereich des Dispatcher kann in verschiedene Kategorien unterteilt werden, z. B. der Koordination der Einheiten zur Behebung von Störungen, der Auftragseinplanung und der Disposition. Bei der Auftragseinplanung werden alle Aufträge, welche von MABI kommen, grob einem Datum zugeordnet. Dabei wird zunächst nur der Hauptvorgang beachtet. Bei der Disposition werden dann der Hauptvorgang und auch die zugehörigen Nebenvorgänge direkt einer Einheit mit Ausführungszeit zugewiesen. Ein einer Einheit zugewiesener Vorgang wird auch als Einsatz bezeichnet. Der Grundprozess der Auftragsausführung ist in Abbildung 2.2 zusammengefasst [3].

Jeder Vorgang besitzt eine Plandauer. Anhand der Plandauer kann der Dispatcher grob abschätzen wie lange der Mitarbeiter für den Vorgang benötigt und somit die Anzahl der abzuarbeitenden Vorgänge pro Tag und Einheit besser planen. Jedoch ist die in RessMa eingepflegte Plandauer eine Schätzung und somit mit Schwankungen behaftet. Je besser die Schätzung der Plandauer ist, desto besser ist das Ergebnis der automatischen Auftragseinplanung und Vordisposition.

Zudem können Einsätze länger als einen Tag dauern, da entweder die Plandauer größer als ein Arbeitstag ist oder ein Einsatz unterbrochen werden musste. Diese heißen dann Mehrtageseinsätze. Für Unterbrechungen gibt es mehrere Ursachen, zum Beispiel kann der Monteur seine Arbeit nicht

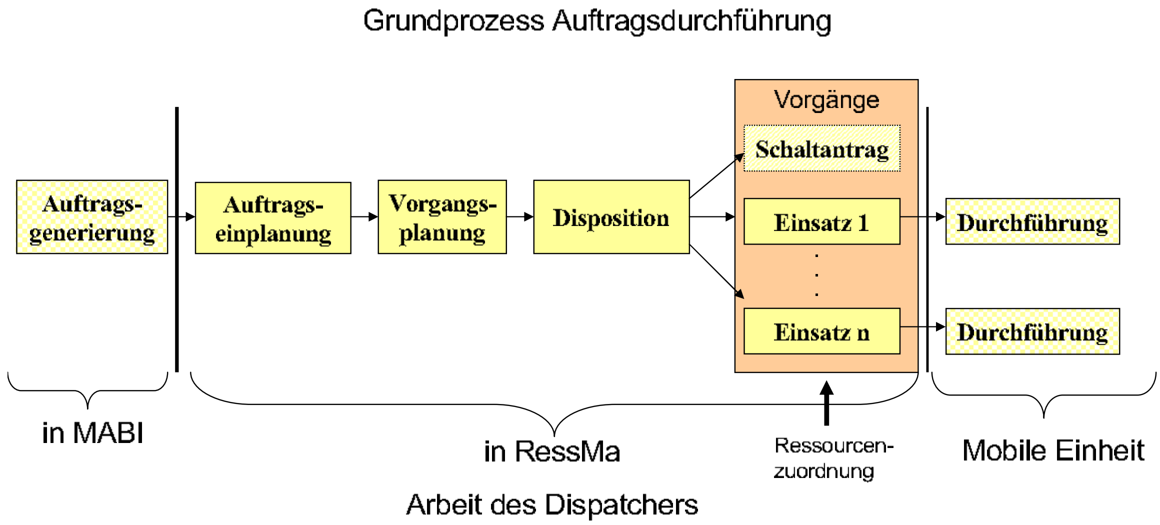


Abbildung 2.2: Grundprozess der Auftragsausführung [3]

beenden, weil er sonst die Arbeitszeit massiv überschreiten würde oder er wird während der Bearbeitung eines Einsatzes plötzlich zu einer Störung gerufen. Diese unterbrochenen Einsätze müssen zu einem späteren Zeitpunkt, aber möglichst zeitnah, beendet werden.

Die gesamte Einsatzdauer setzt sich aus Plandauer und Fahrzeit zusammen. Dabei wird bisher nur die Hinfahrt zum Einsatz berücksichtigt. Die Fahrzeit ist von der Entfernung der Ausführungsorte oder der Startadresse (Wohnort) der Einheit abhängig. Zurzeit wird ein Pauschalwert von ca. $30\,min$ angenommen. Beim letzten Einsatz eines Tages muss die Rückfahrt zum Wohnort bei der Planung mit beachtet werden.

Eine Einheit besteht aus den Ressourcen:

- Mitarbeiter, welcher über Qualifikationen und Berechtigungen verfügt
- Fahrzeug
- Ausrüstung (z. B. Schiebeleiter)

Für jeden Vorgang sind gewisse Ressourcen notwendig. Der Dispatcher wägt ab, ob die gewählte Einheit den Auftrag ausführen kann, auch wenn sie nicht über die geforderten Ressourcen verfügt. Dabei wird unterschieden in Berechtigung, Qualifikation und Ausrüstung. Berechtigungen und Qualifikationen sind meist unabdinglich. Diese Informationen müssen immer auf dem aktuellen Stand sein, da ein Mitarbeiter z. B. durch Weiterbildung zusätzliche Qualifikationen oder durch durch Einweisung weitere Berechtigungen erlangen kann. Bei den Berechtigungen sind vor allem Schaltberechtigungen festgehalten. So kann zum Beispiel nicht jeder Mitarbeiter in jeder Netzstation Schaltungen vornehmen, sondern benötigt dazu die Schaltberechtigung in dieser Netzanlage. Der Grund dafür ist, dass er Anlagenkenntnisse benötigt und das Wissen von Besonderheiten in dieser Anlage. Somit kann eine Einheit nicht im gesamten Netzgebiet der envia NSG für alle Vorgänge eingesetzt werden, sondern besitzt eine gewisse geographische Abgrenzung. Die Anlagen, in denen die Einheiten Berechtigungen besitzt, definiert das bevorzugten Einsatzgebiet.

Die Ausrüstung ist von Einheit zu Einheit unterschiedlich. Es gibt einen gewissen Grundbestand, welchen jede Einheit besitzt. Spezielle Ressourcen, wie zum Beispiel das Kabelschneidgerät, können in der Zentrale abgeholt werden. Jedoch kann nicht jedes Fahrzeug jeden Gegenstand transportieren. Ein normaler PKW kann keine Schiebeleiter transportieren und kann somit Einsätze dieser Art nicht ausführen. Darauf ist beim Dispatchen zu achten.

Im Allgemeinen wird jedem Fahrzeug ein verantwortlicher Mitarbeiter zugewiesen. Dieser fährt im Normalfall das Fahrzeug und ist damit *„Chef der Einheit“*. Nur in Ausnahmefällen, wie zum Beispiel Krankheit, bedient das Fahrzeug ein anderer Mitarbeiter. Jede Einheit verfügt zudem über eine CommBox mit Tablet-PC [1]. In diesen Geräten sind verschiedene Informationen wie z. B. Planwerke hinterlegt, aber auch alle Vorgänge mit den benötigten Beschreibungen, die der Monteur in den nächsten Tagen (meist bis zu 3 Tage) abarbeiten soll. Außerdem dient es als Kommunikationsgerät zwischen Dispachter und Außendienstmitarbeiter und als Navigationsgerät.

Einer Einheit ist eine Organisationsebene mit möglichen Einsatzgebieten zugeordnet. Jedoch sind diese Einsatzgebiete nicht bindend und sie dienen dem Dispatcher ausschließlich der Vorauswahl. Es gibt verschiedene Organisationen, welche zum einem geographisch, aber auch nach Aufgabenbereichen unterschieden werden. Beispiele für Aufgabenbereiche sind MS/NS (Mittel-/Hochspannung) oder SLT (Schutz- und Leittechnik).

Das Versorgungsgebiet der envia NSG unterteilt sich in vier Netzgebiete mit den Hauptstandorten in Markkleeberg für West-Sachsen, Chemnitz für Süd-Sachsen, Cottbus für Brandenburg und Halle für Sachen-Anhalt (siehe Abbildung 2.3). Jedem Netzgebiet ist gleichzeitig ein Dispatchinggebiet zugeordnet. Eine feinere Unterteilung sind die 15 Anlagenmanagement-Gebiete im Bereich der Umspannwerke und des MS/NS-Netzes. Innerhalb dieser Gebiete gibt es wiederum verschiedene Verantwortungsbereiche: Schutz-und Leittechnik, Instandhaltung, Freileitung, Messtechnik und Einsatz NEA/Steiger, woraus sich die unterschiedlichen Organisationen ableiten.

Im System sind zudem Start- bzw. Wohnadresse und die Verfügbarkeit der Einheit hinterlegt. Eine Einheit kann aus verschiedenen Gründen nicht zur Verfügung stehen, so zum Beispiel wegen Urlaub, Krankheit des Mitarbeiters oder Ausfall des Fahrzeuges. Nicht immer ist direkt die Wohnadresse der Einheit abgelegt. Damit eine Versorgung im gesamten Netzgebiet gewährleistet ist, wird stattdessen manchmal eine geeignet zentrale Startadresse gewählte. Ursache dafür ist, dass manche Mitarbeiter nicht im Versorgungsgebiet der envia NSG wohnen oder mehrere in einem kleinen Ballungsgebiet (z. B. in Städten wie Leipzig, wobei Leipzig selbst nicht zum Versorgungsgebiet gehört).

Jeder Auftrag besitzt einen Status, welcher sich vom Status des Hauptvorganges ableitet.

- offen
- vorgeplant
- vordisponiert
- eingeplant

[1] siehe Kapitel Begriffe

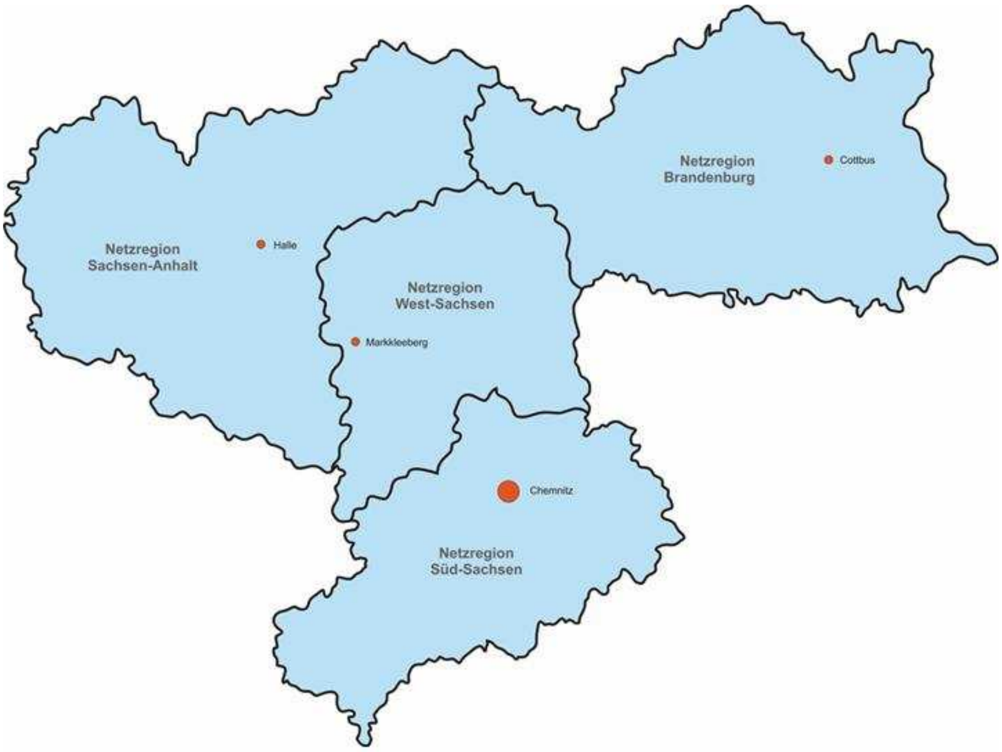


Abbildung 2.3: Netzgebiet der envia NSG

- disponiert
- in Bearbeitung
- operativ erledigt
- erledigt
- storniert

Nicht alle Status sollen an dieser Stelle genau erläutert werden. Es werden nur die wichtigsten herausgegriffen. Ein Auftrag der von MABI kommt, geht als „offen“ oder „vorgeplant“ ein. Werden vom Dispatcher Informationen zu einem Auftrag geändert, so geht der Status in „eingeplant“ über. Verändert der Dispatcher den Status in „disponiert“ um, so wird dieser an die CommBox der gewählten Einheit über GPRS[2] gesendet. Beim Chef der Einheit erscheint der Auftrag auf dem Tablet-PC. Wurde der Hauptvorgang und alle Nebenvorgänge vom Monteur abgearbeitet, so wird dieser Auftrag geschlossen und geht in „erledigt“ über. „Stornierte“ Aufträge wurden nicht abgearbeitet und werden im Folgenden nicht betrachtet.

In den einzelnen Dispatching-Zentralen gehen täglich 200 bis 300 Aufträge ein und einer Dispatching-Zentrale gehören 70 bis 100 Einheiten an. Die Aufträge müssen eingepflegt, vorgeplant und schließlich einer Einheit zugeordnet werden. Die Dispatcher arbeiten dabei an ihren Belastungsgrenzen. Eine Optimierung bezüglich der zurückgelegten Fahrstrecke einer Einheit und der optimalen Auslastung jeder einzelnen Einheit scheint dabei kaum möglich.

Aus diesem Grund entstand das Forschungs- und Entwicklungs-Projekt (F&E-Projekt) „Implementierung einer Optimierungsfunktion in ein WFM-System“, welches zur Aufgabe hat eine Soft-

[2] **G**eneral **P**acket **R**adio **S**ervice bzw. Allgemeiner paketorientierter Funkdienst

ware zu entwickeln, die den Dispatcher unterstützt und eine Optimierung bezüglich der betrieblichen Ziele vornimmt.

Die Voraussetzung bildet das WFM-Teilsysteme RessMa, in dem alle für die Optimierung benötigten Daten, welche bisher genannt wurden und viele weitere, abgebildet sind. Daten, welche noch nicht zur Verfügung standen, aber für die Optimierung erforderlich sind, wurden bis Ende 2009 in RessMa eingepflegt. Darunter zählt das routenfähige Material für eine bessere Fahrzeitminimierung, eine verbesserte Plandauer der einzelnen Vorgänge und die Startadressen der Einheiten.

Das Ziel ist es, am Ende die Arbeit des Dispatchers zu erleichtern und zu optimieren, indem eine automatische Auftragseinplanung und eine automatische Vordisposition vorgenommen wird. Dabei bleibt jedoch der Dispatcher immer derjenige, der Schaltanträge stellt und Vorgänge disponiert. Die Software soll unterstützen, jedoch nicht die Arbeit des Dispatchers ablösen.

Im nächsten Kapitel wird die Aufgabe der Optimierungssoftware im Detail erläutert. Dabei handelt es sich jedoch nicht um eine exakte mathematische Formulierung, sondern zunächst nur um eine Überführung der betrieblichen Ziele als Optimierungsziele.

3 Problemspezifikation

Aufgabe dieses Kapitels ist, die Erläuterung der betrieblichen Ziele und diese als Optimierungsziele bzw. Nebenbedingungen hinsichtlich der Implementierung mit Qualicision® zu interpretieren. Im Folgenden wird das Problem zunächst spezifiziert. Dazu wurde zudem eine Sensitivitätsanalyse mit den Ist-Daten vom Zeitraum 01.07.2008 bis 30.06.2009 durchgeführt. Eine ausführliche Dokumentation ist in [34] zu finden.

3.1 Anforderungen

Das Gesamtziel des F&E-Projektes ist die Kostenminimierung. Diese wird indirekt durch die Senkung des Koordinierungsaufwands im Dispatching und direkt durch die Steigerung der Effizienz des Einsatzes des verfügbaren Arbeitsvolumens für die Bearbeitung des Auftragsvolumens erreicht.

Das Erlangen dieses Dispositionsziels erfolgt zweistufig. Da eine einheitsscharfe Disposition für weit in die Zukunft liegende Tätigkeiten zunächst nicht sinnvoll erscheint (Wochen bis Monate im Voraus), wird in der ersten Stufe eine mittelfristige Auftragseinplanung vorgenommen. In der zweiten Stufe findet dann für die nächsten Tage eine Vordisposition statt (kurzfristig). Im Laufe der Arbeit werden die Zeitdimensionen mittelfristig für Wochen bis Monate und kurzfristig für die nächsten wenigen Tage verwendet.

Automatische Auftragseinplanung Diese beinhaltet die Optimierung der zeitlichen Lage der Aufträge.

Automatische Vordisposition An dieser Stelle erfolgt eine Optimierung der Disposition zeitscharf von Einsätzen auf die Einheiten.

Durch den Einsatz der Optimierungssoftware soll der Koordinierungsaufwand im Dispachting sinken. Damit das Ergebnis der Optimierung auch sinnvoll ist und eine Unterstützung bietet, muss der Prozess der Disposition so genau wie möglich abgebildet werden. Dabei spielen nicht nur die abgeleiteten Bedingungen aus den betrieblichen Vorschriften (z. B. Arbeitszeitregelung) eine Rolle, sondern auch viele weitere Nebenbedingungen, die der Dispatcher bei seiner täglichen Arbeit berücksichtigt (z. B. Gleitzeitregelung). Das Expertenwissen des Dispatchers ist unabdinglich für ein praxistaugliches Ergebnis der Optimierung.

Neben der Unterstützung der Dispatcher soll ebenso eine Effizienzsteigerung erreicht werden. Das Ziel der Effizienzsteigerung lässt sich in fünf betriebliche Ziele (BZ) unterteilen. Die Erläuterung der BZ erfolgt zunächst unter dem Aspekt der Auftragseinplanung und Vordisposition. Im Kapitel 5 erfolgt eine alternative Betrachtungsweise der betrieblichen Ziele.

3.1.1 Betriebliche Ziele (BZ)

BZ 1: Gleichmäßig hohe Auslastung der Organisationen und Einheiten

Bei der Auftragseinplanung soll eine Verteilung der Aufträge so stattfinden, dass alle Organisationen zu 100% ausgelastet sind. Eine Gewährleistung dieses Ziels ist die hohe Anzahl an Aufrägen, d. h. es ist zu jedem Zeitpunkt eine höheres Auftragsvolumen als Arbeitsvolumen vorhanden.

Aufgrund einer höheren Auslastung kann die Vergabe von Aufträgen an Fremdfirmen reduziert und damit der Eigenleistungsanteil erhöht werden. Dies hat die Ursache darin, dass durch effiziente Auslastung der Organisationen Kapazitäten frei werden und Aufträge, welche vorher wegen Kapazitätsmangel fremd vergeben werden mussten, nun wieder in Eigenleistung abgearbeitet werden können.

Bei der Vordisposition wird eine gleichmäßige und hohe Auslastung der Einheiten gewünscht. Dabei ist zu jedem Zeitpunkt auf die Arbeitszeitregelung zu achten. Zudem darf diese Betrachtung nicht nur tageweise, sondern auch wochen- und monatsweise vorgenommen werden, damit die Gleitzeit sinnvoll genutzt wird.

BZ 2: Ressourcenzuordnung

Die Ressourcenzuordnung ist weniger ein Ziel der Kostenminimierung, sondern mehr eins, welches die Zulässigkeit der Disposition gewährleisten soll. Die Beachtung der Qualifikationen und Berechtigungen zwischen Aufträgen und den ausführenden Einheiten dienen vor allem der Einhaltung des Arbeitsschutzes.

Die Ressourcen sind den Einheiten zugeordnet. Da bei der Auftragseinplanung ein Auftrag nur einem Datum und nicht einer Einheit zugeordnet wird, kann die Ressourcenzuordnung erst bei der Vordisposition betrachtet werden.

Die im Kapitel 2 genannten Ressourcen sollen in einem guten Maße berücksichtigt werden. Die Vielfalt und die unterschiedlichen Gewichtungen dieser, bewirken Schwierigkeiten bei der Wahl einer passenden Einheit zu einem Einsatz in der Vordisposition. In den Nebenbedingungen muss genau festgehalten werden, welche Qualifikation, Berechtigung oder Ausrüstung unbedingt Berücksichtigung finden und welche vernachlässigt werden können. Bei Berücksichtigung aller Ressourcen kann es dazu führen, das einige Einsätze keiner Einheit zugeordnet werden können.

BZ 3: Minimierung der Fahrzeit

Die Minimierung der Fahrzeit bewirkt zum einen eine direkte Kostensenkung durch die Reduzierung der Fahrtkosten und zum anderen indirekt durch die Steigung des Arbeitsanteils zum Fahranteil. Durch eine geringere Fahrzeit können mehr Einsätze am Tag bearbeitet werden.

In die Fahrzeit spielen zwei Faktoren hinein. Ein Anteil ist die Fahrt zwischen den einzelnen Einsätzen. Ein weiterer Anteil ist die Hin- bzw. Rückfahrt. Bei der Hinfahrt handelt es sich um den Weg von der Startadresse zum ersten Einsatz und bei der Rückfahrt um den Weg vom letzten Tageseinsatz zur Startadresse. Aus den Ist-Daten wurde eine durchschnittliche Fahrzeit von der Startadresse zum ersten Auftrag von $60\,\mathrm{min}$ ermittelt und zwischen den Einsätzen von $40\,\mathrm{min}$[1].

[1] siehe [34]

Die Fahrzeit für den Rückweg konnte aus den Daten nicht ermittelt werden. Damit wird ersichtlich, dass die Hinfahrt zum ersten Einsatz einen hohen Anteil an der Gesamtfahrzeit pro Tag in Anspruch nimmt. Dabei ist jedoch zu beachten, dass den Werten eine hohe Standardabweichung σ unterliegt (Hinfahrt: $\sigma = 48$ min und zwischen den Einsätzen $\sigma = 37$ min).

BZ 4: Zeitnahes Abarbeiten von Inspektionen in einem Ortsteil

Hintergrund ist, dass die Anlagen eines Ortsteils in einem möglichst kleinen Zeitraum inspiziert werden sollen. Inspektionen i sind Turnusaufträge mit einem Turnus aller j Jahre und einem Ausführungszeitraum $[s_j, e_j]$. Werden Anlagen aus einem Ortsteil zeitlich weit auseinander abgearbeitet, so verschiebt sich auch der Turnus mit dem Ausführungszeitraum. Nach den j Jahren, können die Aufträge dieser Anlagen dann nicht mehr gleichzeitig abgearbeitet werden, da sich die Intervalle der Ausführungszeiträume nicht mehr überschneiden.

In der Auftragseinplanung wird eine Gruppierung der Inspektionsaufträge nach der Anlagenhierarchie vorgenommen. Die Rangfolge dieser Hierarchie lautet

- gleiches Netzobjekt
- gleiches übergeordnete Netzobjekt
- gleicher Ortsteil

Aufträge einer Gruppe sollen dann in ein und dem selben kleinen Zeitraum eingeplant und später vordisponiert werden. Der Begriff zeitnah muss dabei genau definiert werden.

BZ 5: Füllen der Einsatzliste bei bekannten Terminaufträgen

Dieses betriebliche Ziel besitzt eine hohe Gewichtung. Einheiten haben zu ca. 23%[2] mindestens einen Termineinsatz am Tag. Im Mittel besitzen sie dann an diesem Tag $2,1$ Einsätze dieser Priorität. Da Terminaufträge fest und in den meisten Fällen nicht verschiebbar sind, muss die Optimierung passende Fülleinsätze finden. um die Zwischenzeit mit Arbeitszeit zu belegen. Da bei der Auftragseinplanung die Uhrzeit keine Rolle spielt, wird dieses Ziel hauptsächlich in der Vordisposition verfolgt werden.

3.1.2 Größe des Optimierungsgebietes

Neben der genannten betrieblichen Zielen ist festzulegen, wie groß das Optimierungsgebiet sein soll und in welchem Zeithorizont sowohl die Auftragseinplanung als auch die Vordisposition geschehen soll.

Bei der Wahl der Größe des Optimierungsgebietes treten zwei Extreme auf. Auf der einen Seite könnte das gesamte Versorgungsnetz der envia NSG gewählt werden. Dies würde bedeuten, das es keine Grenzen gibt und jeder Monteur eventuell durch das gesamte Gebiet fahren müsste. Dies würde jedoch zu Schwierigkeiten in den Berechtigungen führen, da diese meist nur auf gewisse Anlagen im Vorzugsbereich begrenzt sind. Des Weiteren würden sich die Gebiete der Dispachtingzentralen überschneiden, ein höherer Kommunikationsaufwand zwischen den Dispachern wäre

[2] Der Wert, wie auch alle weiteren, wurde als Durchschnittswerte über ein Jahr aus den Daten der zu verfügungstehenen Datenbank ermittelt und sind nur als Tendenz zu betrachten.

dann gefordert. Ein weiterer Nachteil ist der erhöhte Rechenaufwand, da sowohl alle Aufträge als auch alle Einheiten gleichzeitig optimiert werden müssen. Jedoch ist das Optimierungspotenzial in diesem Fall am höchsten, da es keine Abgrenzungen gibt.

Auf der anderen Seite könnten die kleinste Strukturgröße, die Meisterbezirke für die Optimierung gewählt werden. In einem Meisterbezirk arbeiten ca. 10 bis 15 Mitarbeiter auf einer Fläche von ca. $1000\,km^2$. Auf dieser Ebene ist es bereits möglich eine Optimierung durchzuführen. Jedoch ergibt sich ein höheres Optimierungspotenzial, wenn die Anzahl der zu koordinierenden Einheiten und damit auch die Fläche größer ist.

Zudem sind aus betrieblicher Sicht keine harten Grenzen der Einsatzbereiche der Einheiten gewünscht (überregional). Ein guter Kompromiss zwischen Optimierungspotenzial und Rechenaufwand ist die Wahl der vier Dispachtinggebiete als Optimierungsgebiete.

3.1.3 Zeithorizont der Optimierung

Ein weiterer wichtiger Aspekt ist die Wahl des Zeithorizontes bzw. der Zeithorizonte. Bei Auftragseinplanung und Vordisposition sollten diese unterschiedlich gewählt werden.

Automatische Auftragseinplanung

In der automatischen Auftragseinplanung wird einem Auftrag ein Datum zugeordnet. Das Augenmerk liegt dabei auf dem Hauptvorgang des Auftrages. Alle weiteren zugehörigen Vorgänge werden je nach Relativzeit zum Hauptvorgang im Anschluss eingeplant. Eine Zuordnung einer Einheit geschieht an dieser Stelle noch nicht.

Der Zeithorizont der Auftragseinplanung ist zu beschränken (z. B. auf 90 Tage). Eine Planung über einen zu langen Zeitraum ist nicht sinnvoll, da viele kurz- oder mittelfristige Änderungen, zum Beispiel durch neue Terminaufträge im hinteren Zeitfenster zu vielen Verschiebungen der Aufträge führen würden. Damit wäre das Ergebnis in diesem Zeitbereich nicht robust. Dabei würde ein überflüssiger Ressourcenverbrauch in Zeit und Rechenleistung bei der täglichen Optimierung entstehen.

Alle Aufträge, welche manuell oder bis zum 10. Tag vom System vorgeplant wurden, erhalten eine *Optimierungssperre*, d. h. diese Aufträge werden von der Optimierung in der Auftragseinplanung nicht mehr verschoben. Ursache dafür ist, dass das Zeitfenster von heute bis zum 10. Tage im Blickfeld des Dispatchers liegt. Wurde ein Auftrag bereist manuell vor bzw. eingeplant, dann kann dies Gründe haben, die der Optimierung nicht bekannt sind (z. B. Absprachen) und eine automatische Verschiebung würde die Arbeit des Dispatchers stark behindern. Zudem sollte keinen vollständige Neuplanung von heute auf morgen im Zeitfenster der nächsten 10 Tage stattfinden. Da auch hier der Dispatcher sonst den Überblick verlieren würde.

Aufträge, die am 11. bis zum 90. Tag vom System vorgeplant werden, können durch die Optimierung verschoben werden. Können eingeplante Aufträge nicht ausgeführt werden, dann bleiben dieser zunächst im System als unbearbeitet. In diesem Fall wird der Status des Auftrages wieder in offen überführt und kann damit von der Optimierung neu eingeplant werden.

Automatische Vordisposition

Bei der Vordisposition werden konkret die Vorgänge an die Einheiten der verantwortlichen Organisation zugewiesen. Primär werden die zugehörigen Vorgänge vordisponiert, welche durch die Auftragseinplanung an diesem Tag vorgesehen sind. Können keine passenden Vorgänge aus dieser Menge gefunden werden, so kann zusätzlich auf den gesamten, zur Verfügung stehenden Auftragspool zu gegriffen werden. Neben der Wahl der Einheit wird auch der Termin für den Start des Vorgangs festgelegt (uhrzeitfixe Zuordnung).

Die Vordisposition findet bis zu einem Tag z, betrachtet vom aktuellen Tag x an, statt. Dabei stellt auch hier z ein Parameter dar. Eine Änderung in der bereits vorgenommenen Vordisposition (Zeitraum x bis $x+z$) findet nur dann statt, wenn Terminaufträge bzw. termingebundenen Aufträge mit Ausführungszeitraum x bis $x+z$ noch keiner verfügbaren Einheit zugewiesen sind. Dies können kurzfristige Terminaufträge sein oder Terminaufträge, die von einer plötzlichen Ausfall einer Einheit neu vergeben werden müssen.

Die konkrete Wahl des Parameters z soll mit Hilfe einer späteren Testphase vorgenommen werden. Ein Überblick über die Zeithorizonte gibt die Abbildung 3.1.

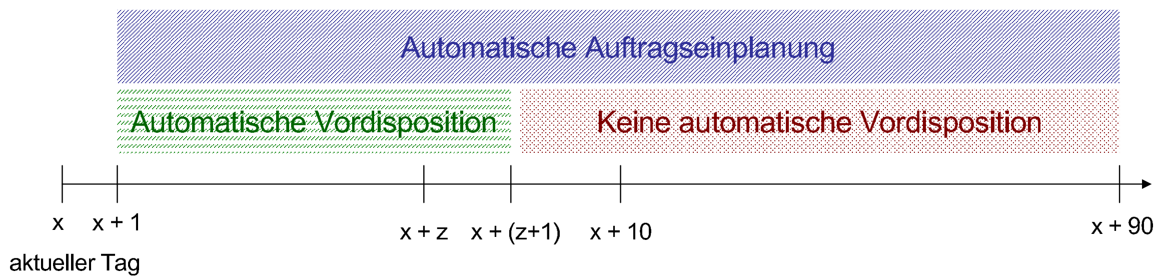


Abbildung 3.1: Zeithorizonte der Auftragseinplanung und Vordisposition mit $z < 10$

3.1.4 Weitere Besonderheiten

Im Gesamten ist der Prozess der Disposition sehr umfangreich. Vor allem die Bestimmung der Zulässigkeit der Einplanung eines Auftrages oder die Disposition eines Vorganges zu einer Einheiten bedarf vieler Kriterien. Die meisten der Kriterien lassen sich in harte oder weiche Nebenbedingungen formulieren (siehe Abschnitt 3.3). Jedoch treten einige Besonderheiten auf, welche an dieser Stelle aufgezeigt werden.

Ein Optimierungspotenzial wird in der Einplanung der Turnus- bzw. nicht priorisierten Aufträge gesehen, also bei den zeitunkritischen Aufträgen. Denn nur diese können frei verschoben und optimal eingeplant werden. Auch die termingebundenen Aufträge bieten ein gewisses Potenzial. Jedoch ist dieses eingeschränkt, da nach der Einplanung diese fix auf einen Tag und Uhrzeit festgelegt sind. Sie können nicht mehr automatisch verschoben werden. Terminaufträge haben eine Optimierungssperre, d. h. sie werden von der Optimierung nicht mehr verändert.

Die Optimierungssperre ist nicht nur für die Einhaltung des Zeithorizontes notwendig, sondern sie ist auch eine Mittel um ungewünschte automatische Änderungen aus der Sicht des Dispatcher zu verhindern. Bereits vom Dispatcher eingeplante oder vordisponierte Aufträge bzw. Vorgänge sollen nicht mehr optimiert werden. Die Optimierungssoftware darf die Arbeit des Dispatchers nicht behindern, sondern sie soll unterstützen. Die Optimierungssperre kann vom Dispatcher jederzeit gesetzt werden. Ursachen dafür kann es verschiedene geben, so zum Beispiel die Absprache mit einem Kunden oder Dienstleister.

Neben Eigenleistung, also Aufträge die von eigenen Mitarbeitern erledigt werden, gibt es Aufträge, die von Fremdfirmen erledigt werden. Da diese nicht mit eigenen Personal bearbeitet werden, erhalten die ebenso eine Optimierungssperre. Dabei ist jedoch zu beachten, dass es Nebenvorgänge des Auftrages geben kann, welche von eigenen Mitarbeitern abgearbeitet werden müssen. Dies ist entsprechend in der Arbeitsauslastung zu berücksichtigen. Ein Beispiel ist die Netzstationreinigung, diese kann von einer Fremdfirma erledigt werden, jedoch sind die Schaltungen vom envia NSG-Personal vorzunehmen. Auch bei Baumaßnahmen mit Überwachung und Absprachen tritt dieser Fall ein.

Im weiteren Verlauf der Arbeit wird der Begriff Optimierungssperre immer dann verwendet, wenn Aufträge oder Vorgänge im Folgenden nicht mehr beachtet bzw. als nicht mehr verschiebbar angenommen werden sollen. Dabei wird darauf hingewiesen, ob die Optimierungssperre nur für die Auftragseinplanung gilt oder auch für die Vordisposition.

Des Weiteren sind einige aus der Praxis bekannte Probleme zu vermeiden. So sollten sich zum Beispiel nicht zeitgleich mehrere Einheiten im selben Ortsteil befinden. Eine Ausnahme ist, wenn mehrere Mitarbeiter für einen Vorgang gefordert sind. Diese werden im Folgenden Sondervorgänge genannt.

Zudem ist es erwünscht und manchmal sogar notwendig, dass Vorgänge eines Auftrages auf die selbe Einheit disponiert werden. Auch sollten begonnene Inspektionen in einem Ortsteil immer von der selben Einheit erledigt werden.

Eine weitere Besonderheit sind Vorgänge, die bereits eine Plandauer größer als acht Stunden haben. Die zugehörigen Einsätze werden auch Mehrtageseinsätze genannt und sollten möglichst zeitnah von der gleichen Einheit abgearbeitet werden.

Es zeigt sich, dass nicht nur das Erreichen der betriebliche Ziele für ein gutes Ergebnis modelliert werden müssen, sondern es ist ebenfalls unabdinglich, dass Ziele aus dem Dispatching und auch Wünsche vom Ausführenden (z. B. Monteur) berücksichtigt werden.

Bei der Modellierung aller genannten Aspekte ist zu unterscheiden, ob sie sich durch Optimierungsziele verwirklichen lassen, mit Hilfe von Nebenbedingungen erreicht werden oder sogar sich durch das Zusammenspiel von selbst mit ergeben. Im nächsten Abschnitt werden zunächst die Optimierungsziele abgeleitet und im Anschluss die verbleibenden Aspekte als Nebenbedingungen formuliert. Dabei ist die Zuordnung eines Kriteriums als Optimierungsfunktion oder als Nebenbedingung nicht immer eindeutig. Auf diesen Sachverhalt wird im nächsten Abschnitt 3.2 eingegangen.

3.2 Überführung der betrieblichen Ziele in Optimierungsziele

Die Auflistung aller betrieblichen Ziele und Besonderheiten wurde bereits im vorangegangenen Abschnitt vorgenommen. Um die betrieblichen Ziele zu erreichen, können diese in Optimierungsziele überführt oder auch als Nebenbedingungen formuliert werden. Bei vielen betrieblichen Ziele ist beides möglich. Innerhalb dieses Abschnittes wird eine Möglichkeit für die Formulierung als Optimierungsziele aufgezeigt und im Anschluss ein paar ausgewählte Nebenbedingungen aufgelistet. Dabei wird zu Grunde gelegt, dass eine später Umsetzung der Optimierung mit Qualicision® erfolgt.

Mit Hilfe der Optimierung sind zwei Aufgaben zu lösen: die eine ist die Auftragseinplanung und die andere die automatische Vordisposition. Mit diesen zwei Teilaufgaben ist die Herangehensweise bereits festgelegt, da beide Aufgaben nacheinander zu lösen sind. Im ersten Schritt werden die Aufträge im Zeithorizont verteilt und im zweiten Schritt dann uhrzeitfix festgelegt und den Einheiten zugeordnet. Somit gelten in der Auftragseinplanung andere Optimierungsziele als bei der Vordisposition. Neben den Optimierungszielen für das Erreichen der betrieblichen Ziele, werden in der Auftragseinplanung Optimierungsziele definiert, welche ein hohes Optimierungspotenzial für die Vordisposition gewährleisten.

BZ 1: Gleichmäßig hohe Auslastung der Organisationen und Einheiten

Auftragseinplanung

Im Laufe des Jahres gehen etwa 150.000 Aufträge mit 214.000 Vorgängen in das System ein. Insgesamt stehen 340 Einheiten im gesamten envia NSG-Gebiet zur Verfügung (siehe [34]). Der tägliche Anteil an zur Verfügung stehenden Einheiten ist jedoch aufgrund von Urlaub und anderen Gegebenheiten geringer. Damit ergibt sich, dass ein hoher Auslastungsgrad für jede Einheit und auch Organisation notwendig ist, um die Bearbeitung der hohen Anzahl an Aufträgen erfüllen zu können.

Dieses Ziel kann damit direkt in ein Optimierungsziel umformuliert werden:

OZ A1: Maximierung der Auslastung der Organisation

Die gleichmäßige Auslastung kann in der Auftragseinplanung durch die Optimierung nicht beeinflusst werden, da die Organisationen nach ihren Qualifikationen bzw. regionalen Begebenheiten geclustert werden und die Zuordnung der Aufträge vom Auftragstyp abhängig ist. Zudem ist das Ziel der gleichmäßigen Auslastung der Organisationen auch nicht erforderlich, da zu jedem Zeitpunkt mehr Aufträge vorliegen als verfügbares Arbeitsvolumen vorhanden ist.

Die automatische Auftragseinplanung wird unabhängig von der Vordisposition durchgeführt. Damit nun jedoch ein für die Vordisposition verwendbare Basis entsteht, müssen weitere Kriterien und Optimierungsziele formuliert werden. Eine Möglichkeit ist es, Auftragspakete zu jeden Tag mit einer Arbeitszeit kleiner gleich acht Stunden zu erstellen. Diese können dann in der Vordisposition den Einheiten zugeordnet werden. Eine Berücksichtigung der Fahrzeit und auch Termine findet in der Auftragseinplanung jedoch nicht statt.

Damit sich nun die Auftragspakete nicht in einem kleinen Gebiet häufen und einzelne Einheiten einen weiteren Anfahrtsweg haben, wird zusätzlich verlangt, dass die Auftragspakete gleichmäßig im Optimierungsgebiet verteilt sind. Ein Beispiel für eine solche Bildung von Auftragspaketen ist in Abbildung 3.2 dargestellt.

OZ A7: Gleichmäßige Verteilung der Auftragspakete

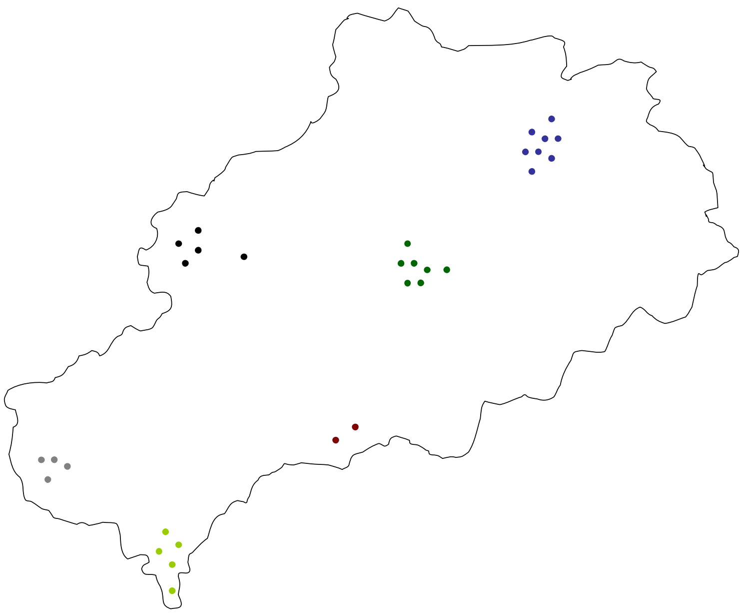


Abbildung 3.2: Gleichmäßige Verteilung der Auftragspakete für einen Tag mit Gesamtplandauer kleiner als acht Stunden

Im Allgemeinen besteht die Gefahr, dass „unpassende“ Aufträge (im Sinne der Optimierung) nicht innerhalb des Ausführungszeitraums im vorgeschriebenen Zeitraum eingeplant und somit manuell bearbeitet werden müssen. Zur Vermeidung dieser Situation wird zusätzlich das Optimierungsziel A6 hinzugenommen.

OZ A6: Minimierung der Anzahl der Aufträge, die nicht im vorgegebenen Ausführungszeitraum eingeplant werden

Da die Optimierungssperre ab dem 11.Tag aufgehoben wird (siehe 3.1.3), können Aufträge ab diesem Tag beliebig verschoben werden. Damit nicht täglich eine komplett neue Auftragseinplanung als Ergebnis entsteht und eine gewisse Konsistenz bewahrt bleibt, wird zusätzlich das Optimierungsziel A8 eingeführt. Dieses Ziel ist ein Beispiel dafür, dass nicht nur die Kostenminimierung verfolgt werden darf, sondern auch die Benutzerfreundlichkeit für den Dispatcher eine entscheidende Rolle spielt.

OZ A8: Minimierung der Verschiebungen von Aufträgen im vorderen Zeitbereich

Vordisposition

Bei der Vordisposition wird nun eine gleichmäßige Aufteilung der von der Auftragseinplanung datumsfix zugeordneten Vorgänge auf die Einheiten gewünscht.

OZ V1: Maximierung der Auslastung der Einheiten und gleichmäßige Verteilung der Vorgänge.

BZ 2: Ressourcenzuordnung

Auftragseinplanung

Bei der Auftragseinplanung muss die Ressourcenzuordnung nicht berücksichtigt werden, da es sich um die Tages- und nicht Einheitenzuordnung handelt.

Vordisposition

Die unter dem betrieblichen Ziel 2 der optimalen Ressourcenzuordnung genannten Forderungen stellen keine Optimierungsziele dar, sondern werden durch Nebenbedingungen in Abschnitt 3.3 berücksichtigt.

Bei der Betrachtung der Ressourcen kann jedoch folgender Aspekt zusätzlich untersucht werden:

OZ V4: Minimaler Ressourcenverbrauch

Dabei sollen Einheiten unter Beachtung der anderen Optimierungsziele bevorzugt werden, welche die benötigten Ressourcen für den Vorgang erfüllen, aber diese nicht zu stark über erfüllen. Dies gewährleistet, dass jede Einheit Aufträge entsprechend ihrer Ressourcen zugeordnet bekommt.

BZ 3: Minimierung der Fahrzeiten

Auftragseinplanung

Da die Zuordnung der Einsätze nach Zeit und Einheit erst bei der Vordisposition vorgenommen wird, kann eine direkte Minimierung der Fahrzeiten nicht vorgenommen werden. Jedoch kann durch geeignete Zuordnung der Aufträge zu den einzelnen Auftragspaketen diese Minimierung indirekt umgesetzt werden.

OZ A2: Minimierung der Mehrfachfahrten zu einem Netzobjekt/Ortsteil

Mit OZ A2 wird erreicht, dass überflüssige Fahrten möglichst vermieden werden und Aufträge innerhalb eines Netzobjektes bzw. Ortsteil an einem Tag erledigt werden. Ausnahmen bilden die Mehrtageseinsätze, in diesem Fall lässt sich ein mehrmaliges Anfahren nicht vermeiden.

Die Auftragspakete werden so erstellt, das die Vordisposition diese im besten Fall dann nur noch den Einheiten zuweisen muss. Aus diesen Grund sollte bereits bei der Auftragseinplanung darauf geachtet werden, dass die Entfernungen der Einsatzorte innerhalb eines Paketes minimal sind (vgl. Abbildung 3.2). Dabei ist als Entfernungsmaß die Luftlinienabmessung ausreichend.

OZ A3: Minimierung der Luftlinienentfernung innerhalb der Auftragspakete

Vordisposition

Innerhalb der Vordisposition kann eine direkte Minimierung der Fahrzeit über die Zuordnung der Aufträge zu den Einheiten nach den Entfernungen und der Reihenfolge der Abarbeitung vorgenommen werden.

OZ V2: Minimierung der Fahrzeiten

Die Fahrzeit wird als linear abhängig zu Strecke angenommen, so dass eine Minimierung der Fahrzeit äquivalent zu Minimierung der Fahrstrecke ist. Um die Fahrzeit zwischen zwei Orten zu erhalten, wird die Fahrstrecke mit Hilfe von routenfähigem Material ermittelt und durch eine Durchschnittsgeschwindigkeit geteilt. Es kann eine Durchschnittsgeschwindigkeit angenommen werden, da die Mehrheit der Fahrten über dass Land und nur selten durch größere Städte führen oder Autobahnen genutzt werden können.

Mit Hilfe einer geringeren Fahrzeit kann ein höherer Anteil an Arbeitszeit im Bezug auf die Fahrzeit erlangt werden. Damit wird gleichzeitig das betriebliche Ziel 1 unterstützt. Dabei spielt auch die Wahl bezüglich der Entfernung des ersten und letzten Einsatzes vom Wohnort für jede Einheit eine entscheidende Rolle.

BZ 4: Zeitnahes Abarbeiten von Inspektionen in einem Ortsteil

Auftragseinplanung

Auch dieses betriebliche Ziel kann direkt in ein Optimierungsziel umformuliert werden.

OZ A4: Minimierung der Zeitspanne der Abarbeitung von Inspektionen in einem Ortsteil

Vordisposition

Bei der Vordisposition gibt es keine Aspekte, die das Erreichen dieses Ziels unterstützen würden.

BZ 5: Füllen der Einsatzliste bei bekannten Terminaufträgen

Auftragseinplanung

Das Füllen von Lücken bei bekannten Terminaufträgen ist mittels Auftragseinplanung nicht möglich, da diese nur datumsfix und nicht uhrzeitfix geschieht. Jedoch kann mit Hilfe der gleichmäßigen Verteilung von Aufträgen mit allgemeiner Terminvorgabe das Optimierungspotenzial für die Vordisposition vergrößert werden. Zum einem entstehen Lücken zwischen den Aufträgen, welche groß genug für die Wahl von Fülleinsätzen sind und zum anderen ist die Wahrscheinlichkeit, dass zwei Aufträge zeitgleich im nahen Umfeld auftreten und somit zwei verschiedene Einheiten diese wahrnehmen müssen, geringer.

OZ A5: Gleichmäßige Verteilung von Aufträgen mit allgemeiner Terminvorgabe

Vordisposition

An dieser Stelle folgt eine direkte Übertragung des betrieblichen Ziels in das Optimierungsziel:

OZ V3: Minimierung der nicht belegten Arbeitszeit durch Schließen der Lücke zwischen den Terminaufträgen

3.3 Nebenbedingungen

Das Problem der optimalen Einsatzplanung ist sehr komplex und fordert somit eine Fülle an Nebenbedingungen. All diese Nebenbedingungen werden an dieser Stelle nicht erläutert und können in [2] nachgelesen werden. Es werden nur ein paar Wichtige herausgegriffen und näher beschrieben.

Optimierungssperre Der Begriff der Optimierungssperre trat bereits an verschiedenen Stellen auf. Es gibt verschiedene Ursachen dafür, dass ein Auftrag eine Sperre für die Optimierung erhält. Automatisch wird diese in der Auftragseinplanung zum Beispiel gesetzt, wenn ein eingeplanter Auftrag in den Zeitbereich aktueller Tag plus 10 Tage gelangt oder wenn der Auftrag nicht mit eigenen Personal bearbeitet wird (Fremdleistung). Auch manuell kann der Auftrag für die Optimierung gesetzt werden. Dies geschieht vom Dispatcher.

Egal aus welchen Grund die Optimierungssperre gesetzt wird, die Optimierungssoftware muss beachten, dass diese Aufträge nicht automatisch eingeplant oder verschoben werden dürfen. Jedoch müssen bereits eingeplante Aufträge in der Berechnung der Auslastung berücksichtigt werden. Eine Ausnahme bilden die Fremdleistungen. Die Optimierungssperre gilt nur für den Hauptvorgang, d. h. die Nebenvorgänge können durch die Optimierung im vorgesehenen Zeitraum eingeplant werden. Auch hier gibt es die Ausnahme bei Nebenvorgängen, welche durch Fremdfirmen abgeleistet werden. Ein Beispiel ist die Verteilung der Abschaltinformation, welche bei großen Gebieten oft von Fremdfirmen erledigt werden.

Im Allgemeinen gilt die Optimierungssperre zunächst nur für die Auftragseinplanung. Soll diese auch für die Vordisposition gelten, dann muss extra darauf hingewiesen werden. So kann es zum

Beispiel sein, dass der Dispachter für einen Auftrag eine spezielle Einheit vorgesehen hat, dann kann er diese entweder fest disponieren, so dass die Optimierung auf diesen Auftrag nicht mehr zugreifen kann oder er setzt die Optimierungssperre bei der Vordisposition.

Zudem ist immer zu beachten, dass wenn ein Hauptvorgang, egal ob von der Auftragseinplanung oder von der Vordisposition verschoben wird, dann müssen auch alle Nebenvorgänge mit verschoben werden, wenn dies notwendig ist. Die Nebenvorgänge müssen immer in der vorgesehenen Relativzeit eingeplant werden.

Arbeitszeit und Gleitzeit Gesetzlich vorgeschrieben ist die Einhaltung der maximalen Arbeitszeit pro Tag und pro Woche. Diese Grenzen sind unbedingt einzuhalten. Zudem gibt es die von der Firma festgelegte Wochenarbeitszeit, welche durch die Gleitzeit eine kleine Spannweite hat. Bei envia NSG beträgt die Wochenarbeitszeit 38 Stunden, das ergibt eine Tagesarbeitszeit von 7,6 Stunden. Diese kann zwischen 5 und 10 Stunden variieren. Zunächst wird der Einfachheit halber eine Tagesarbeitszeit von 8 Stunden angenommen. Auch die Wochenarbeitszeit muss nicht korrekt eingehalten werden. Wichtig ist, dass im Mittel keine Unter- oder Überstunden anfallen. Dies ist eine sehr weiche Beschreibung der Nebenbedingungen für die Arbeitszeit.

Da der Kern der Software, welche zum Einsatz kommt, auf Fuzzy Optimierung beruht, können die Nebenbedingungen, welche die Arbeitszeit betreffen weich formuliert werden. In der Abbildung 3.3 ist eine mögliche Kurve für die Tagesauslastung aufgezeigt. Interpretiert wird dies wie folgt: eine Arbeitszeit von 8 Stunden wird angestrebt. Die Gewichtung ist 1. Eine Überschreitung von einer Stunde erhält noch ein Gewichtung von 0.9 und ist damit zulässig. Auch eine Über- bzw. Unterschreitung von 2 Stunden ist zulässig. Die Gewichtung ist jedoch nur noch 0,5 und damit nicht mehr angestrebt. Ab eine Abweichung von 3 Stunden zu den 8 Stunden ist die Gewichtung 0 und damit nicht zulässig. Diese Kurve ist eine erste Annahme und muss im Laufe der Testphase angepasst werden.

Ressourcen Die Ressourcen Ausrüstung, Berechtigungen und Qualifikationen sind sehr vielfältig. Doch bei genauerer Betrachtung des Vorkommen der Ressourcen während der Sensitivitätsanalyse (siehe [34]) wurde deutlich, dass nur gute 40 verschiedene Ressourcen in der Datenbank aufgelistet wurden. Jedoch ist anhand der Häufigkeit des Vorkommens einer Ressource nicht die Wichtigkeit dieser zu Erkennen. Der Dispatcher muss an dieser Stelle sein Wissen einbringen. Es muss entschieden werden, welche Ressourcen unabdinglich sind und welche zu vernachlässigen sind.

Eine scharfe Einhaltung aller Ressourcen in den Nebenbedingungen kann zu einer leeren Lösungsmenge führen. Die Optimierung kann somit keine Lösung liefern. Um dies zu vermeiden werden nur die wichtigsten Ressourcen herausgefiltert und der Optimierung übergeben. Alle anderen werden zunächst vernachlässigt. In Spezialfällen muss der Dispatcher entscheiden, ob eine Einheit einen dieses Auftrag ausführen darf bzw. kann.

Bemerkungen zu den Nebenbedingungen Es wurden nur einige Nebenbedingungen aufgelistet, aber bereits an diesen wird ersichtlich, dass das Finden einer zulässigen Lösung, d. h. eine Lösung, welche alle Nebenbedingungen erfüllt, bereits schwierig ist. Bei der Umsetzung in der Optimierungssoftware muss beachtet werden, dass die Einschränkungen nicht zu groß sind, da sonst eine Optimierung kaum noch möglich ist. Jede Nebenbedingung sollte aus diesem Grund auf ihre

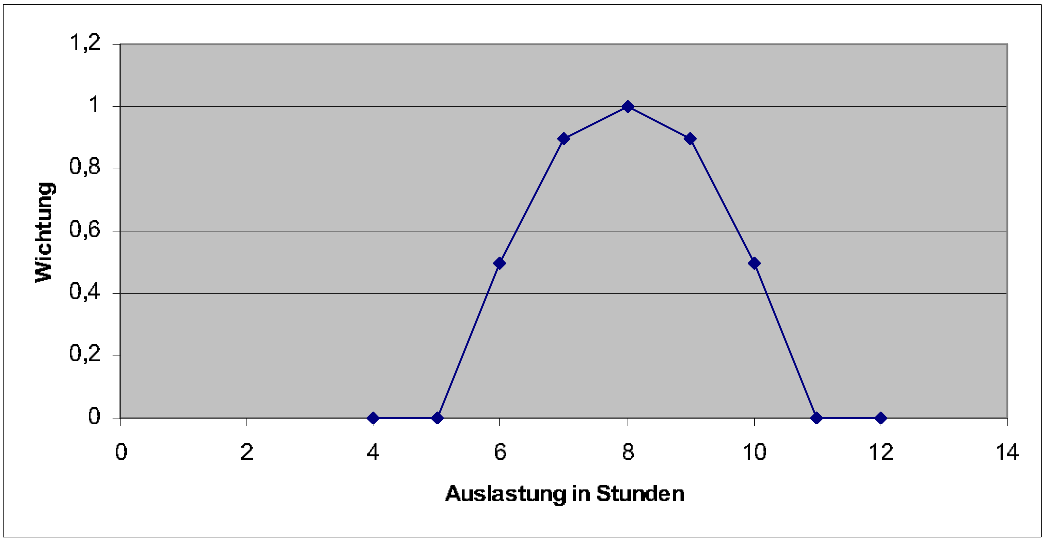


Abbildung 3.3: Gewichtung der Arbeitszeit einer Einheit pro Tag

Notwendigkeit geprüft werden. Zudem vergrößert sich das Problem je mehr Nebenbedingungen hinzukommen. Somit wächst zusätzlich der Zeitaufwand zur Berechnung.

Eine weitere Besonderheit sind die zum Beispiel bei der Arbeitszeit auftretenden weichen Nebenbedingungen, d. h. es werden keine harten Grenzen für die Arbeitszeit gegeben, sondern weiche Übergänge. Ein Beispiel ist in der Abbildung 3.3 als Funktion dargestellt. Dies hat den Vorteil, dass die zulässige Lösungsmenge vergrößert wird. Zur Lösung von Optimierungsaufgaben mit weichen Nebenbedingungen kann die Fuzzy-Optimierung herangezogen werden.

3.4 Zusammenfassung

In diesem Kapitel wurden das Problem der Disposition in zwei Teilaufgaben geteilt: der Auftragseinplanung und der Vordisposition. Beide Aufgaben können getrennt voneinander abgearbeitet werden. Dabei ist es erforderlich zunächst die Auftragseinplanung zu lösen und im Anschluss die Vordisposition vorzunehmen.

Sowohl die Auftragseinplanung als auch die Vordisposition beinhalten mehrere Optimierungsziele. Somit sind zwei Polyoptimierungsaufgaben durch die Software zu lösen. Die in diesem Kapitel vorgenommen Problemspezifikation geschah immer im Hinblick auf die Lösung mit Hilfe der Fuzzy-Optimierung. Somit konnten zum Beispiel die Nebenbedingungen weich formuliert werden. Eine weiterführende Beschreibung der Umsetzung des Problems, die Auflistung weiterer Sonderfälle und der Umgang mit diesen ist dem Pflichtenheft [2] zu entnehmen.

Somit wurde nun eine Einblick in das gegeben Problem und ein Ansatz zu Modellierung der Auftragseinplanung und Vordisposition gegeben. Im Folgenden besteht nun das Ziel, das Problem aus einem anderen Blickwinkel zu betrachten und ein alternativen Lösungsansatz zu schaffen. Bevor solch einer jedoch ausführlich beschrieben wird, werden einige mathematische Grundlagen benötigt. Diese werden im nächsten Kapitel erläutert.

4 Mathematische Grundlagen und Methoden der Optimierung

4.1 Lineare und diskrete Optimierung

Die Optimierung ist ein breites Gebiet in der Mathematik. Es wird vor allem in die lineare, nichtlineare und diskrete Optimierung unterschieden. Innerhalb dieser Arbeit werden die Grundlagen der linearen und diskreten Optimierung benötigt. Die wichtigsten Begriffe werden im Folgenden wiederholt. Eine detaillierte Beschreibung kann unter anderem in [8] und [12] nachgeschlagen werden.

Die Optimierung dient der Entscheidungsfindung bei komplexen Problemen. Zunächst wird ein Modell aufgestellt, welches eine oder mehrere Zielfunktionen und einige Bedingungen beinhaltet. Dieses Modell soll das Problem mathematisch widerspiegeln. Existieren mehrere Zielfunktionen, so gehört das Problem der *Polyoptimierung* an. Im weiteren Verlauf werden jedoch ausschließlich Optimierungsprobleme mit nur einer Zielfunktion betrachtet. Die Bedingungen, unter denen die Optimierung geschehen soll, werden Nebenbedingungen oder Restriktionen genannt. Bei einer linearen Optimierungsaufgabe (LOA) sind sowohl die Zielfunktion, als auch die Nebenbedingungen linear.

Definition 3 (lineare Optimierungsaufgabe)**.** *Als lineare Optimierungsaufgabe (LOA) wird das Problem der Ermittlung aller Vektoren* $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^T$ *bezeichnet, die unter Beachtung von linearen Nebenbedingungen (NB) und zumeist unter Berücksichtigung von Nichtnegativitätsbedingungen (NNB) eine lineare Zielfunktion optimieren. Eine lineare Optimierungsaufgabe lässt sich wie folgt formulieren:*

(LOA)

$$\text{ZF:} \qquad \boldsymbol{c}^T \boldsymbol{x} \rightarrow opt. \tag{4.1a}$$

$$\text{unter den NB:} \qquad A\boldsymbol{x} \leq \boldsymbol{b}\,, \tag{4.1b}$$

$$x_i \geq 0 \qquad \forall i \tag{4.1c}$$

Die erste Gleichung spiegelt die Zielfunktion wider und $\boldsymbol{c}$ wird als Zielfunktionsvektor bezeichnet. Eine Besonderheit in der linearen Optimierung ist, dass sich Gleichungen bzw. Ungleichungen der Nebenbedingungen in Matrizengleichungen bzw. -ungleichungen umschreiben lassen. Die Matrix A aus dem Modell (4.1) umfasst alle Nebenbedingungen. Jede Art von linearen Nebenbedingungen können in diese Form gewandelt werden. So können zum Beispiel auch Gleichungsnebenbedingungen durch äquivalenten Umformungen in Ungleichungsnebenbedingungen umgewandelt werden [8]. Die durch die Nebenbedingungen einschließlich der Nichtnegativitätsbedingungen definierte Menge wird als *Restriktionsbereich* bezeichnet. Eine Lösung, welche innerhalb des Restriktionsbereich liegt, d. h. für die alle Nebenbedingungen (NB und NNB) erfüllt sind, heißt zulässig. Bei Maximierungsproblemen (Minimierungsproblemen) ist diejenige zulässige Lösung optimal, welche den größten (kleinsten) Zielfunktionswert besitzt. Eine optimale Lösung wird im Laufe der Arbeit mit $\boldsymbol{x}^*$ bezeichnet.

Die Anwendung der LOA ist sehr vielfältig. Beispiele sind die Produktionsplanung oder das Zuschnittsproblem. Zur Lösung von LOA gibt es ebenso verschiedene Herangehensweisen. Die bekannteste und meist verwendetste Methode ist die Simplexmethode. Dies ist ein numerisches Verfahren, dass nach endlich vielen Schritten die optimale Lösung liefert bzw. die Information, dass die Lösungsmenge leer ist oder der Restriktionsbereich in Optimierungsrichtung nicht beschränkt ist. Eine ausführliche Beschreibung dieser Methode ist in [9] zu finden.

Eine spezielle Form der linearen Optimierungsprobleme sind solche mit zusätzlicher Ganzzahligkeitsforderung. Diese werden auch als ganzzahlige LOA (GLOA) bezeichnet. Sie sind ein Teil der diskreten Optimierung. Bei diskreten Optimierungsaufgaben können alle oder ein Teil der Variablen nur Werte aus einer diskreten Mengen annehmen.

Definition 4 (diskrete Optimierungsaufgabe)**.** *Eine diskrete Optimierungsaufgabe (DOA) ist eine lineare oder nichtlineare Optimierungsaufgabe, deren Restriktionsbereich ein diskrete Menge ist.*

(DOA)

$$\textit{ZF:} \qquad f(\boldsymbol{x}) \rightarrow opt. \tag{4.2a}$$

$$\textit{unter den NB:} \qquad \boldsymbol{x} \in R_D \tag{4.2b}$$

Dabei ist R_D eine diskrete Menge. Eine diskrete Menge ist entweder eine endliche Menge oder besteht ausschließlich aus isolierten Punkten[1].

Besteht R_D aus den nichtleeren Mengen D und G ($R_D = D \times G$), wobei G eine beliebige Menge und die Menge D diskret sind, dann wird sie *gemischt diskret* genannt. Das zugehörige Optimierungsproblem wird ebenso als *gemischt diskretes Optimierungsaufgabe* bezeichnet.

Die diskreten Optimierungsaufgaben lassen sich wiederum klassifizieren. Zum Beispiel in:

Ganzzahlige Optimierungsaufgabe (GOA)

$$\text{ZF:} \qquad f(\boldsymbol{x}) \rightarrow opt. \tag{4.3}$$

$$\text{unter den NB:} \qquad \boldsymbol{x} \in R \tag{4.4}$$

$$x_j \in \mathbb{Z} \qquad j = 1, \ldots, n \tag{4.5}$$

[1]Ein Punkt p einer Menge S wird als isolierter Punkt bezeichnet, wenn es eine Umgebung $K(p, \epsilon)$ gibt, so dass kein weiteres Element aus S in dieser Umgebung liegt $(K(p, \epsilon) \cap S = \{p\})$. Andernfalls wird p auch Häufungspunkt genannt.

Ganzzahlige lineare Optimierungsaufgabe (GLOA)

$$\text{ZF:} \qquad \boldsymbol{c}^T\boldsymbol{x} \rightarrow opt \tag{4.6}$$

$$\text{unter den NB:} \qquad A\boldsymbol{x} \leq \boldsymbol{b} \tag{4.7}$$

$$x_j \geq 0,\ x_j \in \mathbb{Z} \qquad j = 1, \ldots, n \tag{4.8}$$

Kombinatorische OA: R_D in (4.2) ist eine endliche Menge

Boolsche/binäre OA: x_j kann nur die Werte 0 und 1 annehmen ($j = 1, \ldots, n$)

Ein Optimierungsproblem kann verschiedenen Klassen gleichzeitig angehören. So sind kombinatorische Optimierungsaufgaben meist Spezialfälle der ganzzahligen linearen Optimierungsaufgaben und die binäre OA ist wiederum ein Spezialfall der kombinatorischen.

Im Rahmen dieser Arbeit spielen vor allem kombinatorische Optimierungsproblem eine Rolle. Auch wenn es zunächst den Anschein macht, dass diese Optimierungsprobleme leichter zu lösen sind, so ist dies nicht der Fall. Zwar ist die Menge der zulässigen Lösungen endlich, jedoch kann die Mächtigkeit dieser Menge so hoch sein, dass eine vollständige Enumeration[2] in angemessener Zeit nicht möglich ist.

In der diskreten Optimierung sind die bekannten und effizienten Verfahren aus der linearen Optimierung nicht anwendbar. Somit bedarf es neuer Verfahren. Die Schwierigkeit in der diskreten Optimierung ist, dass es nicht ein Verfahren gibt, welches sich auf alle anwenden lässt. Die Vielfalt der Verfahren ist beinahe so groß wie die Vielfalt der Probleme. Jedoch gibt es allgemeine Herangehensweisen, welche immer an das jeweilige Problem angepasst werden müssen. Im folgenden Abschnitt werden verschiedene allgemeine Lösungsmethoden vorgestellt.

4.2 Allgemeine Lösungsmethoden in der diskreten Optimierung

Bei den Lösungsverfahren in der diskreten Optimierung wird in exakten Verfahren und Näherungsverfahren unterschieden. Die exakten Verfahren liefern nach endlich vielen Schritten eine optimale Lösung bzw. die Erkenntnis, dass keine existiert. Darunter zählen die Simplexmethode, die Schnittebenenverfahren, die Entscheidungsbaumverfahren und die vollständige bzw. unvollständige Enumeration. Im Gegenzug liefern die Näherungsverfahren ein zulässige und nur *fast optimale* Lösung. Fast optimale Lösungen sind Lösungen, die nahe am Optimum liegen oder sogar gleich dem Optimum sind, jedoch erkennt das Verfahren nicht, dass bereits eine optimale Lösung gefunden wurde. Typische Näherungsverfahren sind unvollständige exakte Verfahren und Eröffnungs- mit Verbesserungsverfahren.

In der Praxis ist die Ausführung exakter Verfahren meist nicht möglich. Ursache dafür sind die Komplexität der Probleme und der damit verbundene Rechenaufwand. Oft sind zudem die Eingangsdaten mit einer Ungenauigkeit bzw. Unsicherheit behaftet. Damit ist es meist auch nicht notwendig, *die* optimale Lösung zu erhalten, sondern es wird nach einer zulässigen Lösung gefordert, welche nahe dem Optimum liegt. Aus diesem Grund spielen die Näherungsverfahren in der

[2] Bezeichnung für folgendes Vorgehen: Berechnung des Zielfunktionswertes für alle zulässigen Lösungen und Herausfiltern der optimalen Lösung(en).

Praxis eine wesentliche Rolle. Dabei muss oft ein Kompromiss zwischen Güte der Lösung und dem Aufwand der Ermittlung dieser Lösung eingegangen werden.

Bei den unvollständigen exakten Verfahren wird abgebrochen bevor die optimale Lösung ermittelt wurde, bzw. bevor bekannt wurde, dass diese bereits erreicht ist. Ein weit verbreitetes exaktes Verfahren ist das Entscheidungsbaum-Verfahren Branch&Bound. Dieses Verfahren kann vorzeitig abgebrochen werden, da es zu jedem Zeitpunkt zulässige Lösungen liefert.

Eröffnungsverfahren ermitteln eine zulässige Lösung, diese wird im Anschluss durch die Verbesserungsverfahren aufgegriffen. Mit Hilfe des Verbesserungsverfahren wird durch sukzsessive Änderungen der Lösung ein lokales Optimum erreicht. Dabei ist es vom Aufwand und der Güte des Eröffnungsverfahren abhängig, wie viel Aufwand im Anschluss in das Verbesserungsverfahren noch eingebracht werden muss.

Ist das vorgelegte diskrete Optimierungsproblem eine ganzzahlige lineare Optimierungsaufgabe so kann ein Zusammenhang zu den Lösungsverfahren der LOA hergestellt werden. Die Lösung des diskreten Problems erfolgt in zwei Schritten. Zunächst werden die Ganzzahligkeitsforderung $\boldsymbol{x} \in \mathbb{Z}$, und wenn nötig weitere Nebenbedingungen, weggelassen. Es entsteht ein relaxiertes Problem.

Definition 5 (Relaxation)**.** *Ein relaxiertes Problem ist ein Problem, welches aus einer Optimierungsaufgabe durch Weglassen einer oder mehrere Nebenbedingungen entsteht.*

Durch die Relaxation der Ganzzahligkeitsforderung wird aus der GLOA eine lineare Optimierungsaufgabe. Diese kann nun mit den bekannten Methoden aus der linearen Optimierung gelöst werden. Das Ergebnis ist jedoch in den meisten Fällen nicht ganzzahlig und damit nicht zulässig für die Ausgangsaufgabe. Doch aufgrund des folgenden Satzes ist der Zielfunktionswert der ermittelten Lösung zunächst eine gute obere bei Maximierungs- bzw. eine gute untere Schranke bei Minimierungsproblemen.

Satz 1. *[7] Das relaxierte Problem zu einer Optimierungsaufgabe bei der der Zielfunktionswert zu maximieren (minimieren) ist, hat einen optimalen Zielfunktionswert, welcher größer (kleiner) oder gleich dem der Ausgangsaufgabe ist.*

Die linear zulässige Lösung muss im Anschluss durch weitere Verfahren so modifiziert werden, dass sie auch für die diskrete Ausgangsaufgabe zulässig ist. Ein einfaches Runden der Werte führt meistens nicht zu einer akzeptablen Lösung. Jedoch kann diese Lösung als Ausgangspunkt für Verbesserungsverfahren verwendet werden. Eine optimale Lösung des relaxierten Problems wird im Bezug auf die Ausgangsaufgabe als *fast zulässig* bezeichnet.

4.3 Beispiele wichtiger Optimierungsaufgaben

Im letzten Abschnitt wurden kurz die allgemeinen Methoden zur Lösung diskreter Optimierungsaufgaben vorgestellt. In diesem Abschnitt wird nun speziell auf einige konkrete Optimierungsaufgaben eingegangen. Zunächst wird das Problem beschrieben und eine mögliche Modellierung angegeben. Im Anschluss werden typische Lösungsverfahren dargelegt. Es wird vor allem auf Beispiele eingegangen, welche im Laufe der Arbeit noch benötigt werden.

4.3.1 Transportprobleme

Das klassische Transportproblem ist ein spezielles lineares Optimierungsproblem. Es existieren m Ausgangsorte A_i $(i = 1, \ldots, m)$ mit der Lagerung der Mengen a_i eines Produktes und n Bestimmungsorte B_j $(j = 1, \ldots, n)$ mit dem Bedarf b_j an diesen Produkt. Zudem sind die Kosten c_{ij}, welche beim Transport einer Mengeneinheit von A_i nach B_j entstehen, bekannt. Diese können zum Beispiel die Entfernungen sein oder der Zeitaufwand bzw. die Kosten für den Transport.

Die Aufgabe besteht nun darin einen Transportplan $X = (x_{ij})_{m,n}$ mit den Transportmengen x_{ij} von A_i nach B_j zu erstellen, so dass die Transportkosten minimal sind. Zudem ist beim klassischen Transportproblem gefordert, dass der Gesamtbedarf gleich der gesamten Auslieferungsmenge ist, also:

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j \tag{4.9}$$

Zusammenfassen lassen sich diese Informationen im folgenden mathematischen Modell:

(TP)

$$\text{ZF:} \quad \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \rightarrow min \tag{4.10a}$$

$$\text{unter den NB:} \quad \sum_{j=1}^{n} x_{ij} = a_i \qquad i = 1, \ldots, m \tag{4.10b}$$

$$\sum_{i=1}^{m} x_{ij} = b_j \qquad j = 1, \ldots, n \tag{4.10c}$$

$$x_{ij} \geq 0 \qquad i = 1, \ldots, m; \quad j = 1, \ldots, n \tag{4.10d}$$

Die Zielfunktion (4.10a) minimiert die Gesamttransportkosten. Die Nebenbedingung (4.10b) gewährt, dass die gesamten Angebotsmengen aus den Lagern A_i abtransportiert werden und die Nebenbedingung (4.10c), dass der Bedarf aller Orte B_j gedeckt wird.

Jede Lösung, die alle Nebenbedingungen erfüllt, ist zulässig und stellt einen Transportplan dar. Alle zulässigen Lösungen bilden ein Polyeder, welches auch Transportpolyeder genannt wird. Das Transportpolyeder T lässt sich wie folgt beschreiben:

$$T = \{X \in \mathbb{R}^{m \times n} | \sum_{i=1}^{m} x_{ij} = a_i, \sum_{j=1}^{n} x_{ij} = b_j, x_{ij} \geq 0, \forall i = 1, \ldots, n, j = 1, \ldots, m\}$$

Jede eindeutige optimale Lösung entspricht einer Ecke dieses Polyeders T. Liegt eine mehrfache Lösung vor, so ist dies eine Kante bzw. Hyperebene des Polyeders. Eine optimale Lösung ist zudem eine *Basislösung* [8]. Diese hat die Eigenschaft, dass die Spalten von A mit positiven Variablen x_{ij} linear unabhängig sind.

Das klassische Transportproblem ist ein spezielles lineares Optimierungsproblem mit $m + n$ Ungleichungsnebenbedingungen und $m \times n$ Variablen. Die Restriktionsmatrix A besteht ausschließlich aus Nullen und Einsen. Zudem ist die Spaltensumme 2. Des Weiteren ergibt sich durch die Bedingung (4.9) die lineare Abhängigkeit der Nebenbedingungsgleichungen. Der Rang der Matrix A ist somit $r(A) = n + m - 1$.

Die Anwendung des Simplex-Verfahrens ist möglich, wäre jedoch ineffizient, da A bereits durch die Dimensionen m und n eindeutig bestimmt ist und die Anzahl der Variablen, welche zusätzlich eingeführt werden müssten, sehr hoch ist [9]. Aus diesem Grund wurden weitere Algorithmen zu Lösung des Transportproblems entwickelt. An dieser Stelle finden die Eröffnungs- und Verbesserungsverfahren Anwendung.

Eröffnungsverfahren für das klassische Transportproblem sind zum Beispiel die Nord-West-Ecken-Regel, die Methode des kleinsten Elementes oder die Vogelsche Approximation. Die Nord-West-Ecken-Regel ist ein sehr einfaches Verfahren, welche eine zulässig Lösung ermittelt. Jedoch ist dies nur irgendeine zulässige Lösung. Die beiden weiteren Verfahren berücksichtigen die Zielfunktionskoeffizienten, also die Kosten und sind aus diesem Grund zwar aufwendiger, aber das Ergebnis besitzt einen besseren Zielfunktionswert. Die Vogelsche Approximation liefert in vielen Fällen bereits die optimale Lösung. Der Algorithmus der Vogelschen Approximation wird im Abschnitt 4.4.1 dargelegt.

Als Verbesserungsverfahren sei an dieser Stelle die Potentialmethode[3] genannt. Sie stützt sich auf der besonderen Gestalt des Tableaus bzw. der Dualitätstheorie[4] und liefert nach endliche vielen Schritten die optimale Lösung. Auch dieser Algorithmus wird im Abschnitt 4.4.2 angegeben.

In der Praxis liegt in den wenigsten Fällen das klassische Transportproblem vor. Oft sind weitere Bedingungen gefordert oder die Voraussetzung (4.9) ist nicht erfüllt. Typische Verallgemeinerungen sind:

a) Gleichung (4.9) ist nicht erfüllt: Zum einen könnte das Angebot größer als der Bedarf sein und zum anderen der Bedarf größer als das Angebot. Im ersten Fall wird ein fiktiver Bedarfsort b_{n+1} mit dem überschüssigen Angebot eingeführt:

$$b_{n+1} = \sum_{i=1}^{m} a_i - \sum_{j=1}^{n} b_j \tag{4.11}$$

   Analog werden im zweiten Fall fiktive Angebotsorte eingeführt. Die Transportkosten zu den fiktiven Orten werden gleich Null gesetzt. Mit Hilfe dieser Einführung der fiktiven Bedarfs- bzw. Angebotsorte kann die Gleichung (4.9) wieder erfüllt werden.

b) Einige der Wege sind gesperrt: Ist die Verbindung von A_i nach B_j nicht passierbar oder aus anderen Gründen unerwünscht, dann werden die Kosten $c_{ij} = M_\infty$ gesetzt, wobei M_∞ eine hinreichend groß Zahl ist, z. B. $M_\infty = \sum_{j=1}^{m} \sum_{i=1}^{n} c_{ij}$.

c) Es existieren Liefer- oder Annahmebedingungen, d. h. x_{ij} soll größer einer gewissen unteren Schranke u_{ij} für gewisse (i, j) sein ($x_{ij} \geq u_{ij}$). Untere Schranken können vorab mit berücksichtigt werden. Dazu wird $\widetilde{a_i} = a_i - \sum_{j=1}^{m} u_{ij}$ und $\widetilde{b_j} = b_j - \sum_{i=1}^{n} u_{ij}\ \forall i, j$ gebildet. Es ergibt sich ein reduziertes Transportproblem und zur optimalen Lösung X^* müssen die Schranke im Anschluss wieder addiert werden: $x^*_{ij} = \widetilde{x}^*_{ij} + u_{ij}$.

Mit den beschriebenen Modifikationen lassen sich die verallgemeinerten Transportprobleme wieder in das klassische überführen und dann mit den oben genannten Methoden lösen.

[3] In der Literatur auch unter dem Namen MODI-Methode oder Transportmethode bekannt.
[4] siehe Literatur [6]

Ein weiteres spezielles Transportproblem ergibt sich, wenn $a_i, b_j \in \mathbb{N}\ \forall i, j$ gilt. In diesem Fall kann das Problem der diskreten Optimierungsaufgabe zugeordnet werden, da damit $x_{ij} \in \mathbb{N}\ \forall i, j$ gilt. Die Ursache dafür ist, dass die Verfahren zur Ermittlung der optimalen Lösung nur die Operationen: Addition, Multiplikation und Minimum-Bildung auf den natürlichen Werte a_i, b_j verwenden (vgl. Abschnitt 4.4.1 und 4.4.2). Somit kann durch Multiplikation eines geeigneten Faktors jedes linearen Transportproblem in ein diskretes Problem umgewandelt werden.

4.3.2 Lineare Zuordnungsprobleme

Das Zuordnungsproblem ist ein spezielles Transportproblem. Dabei wird jedes Objekt j genau einem Mittel i und umgekehrt zugeordnet. Für die Lösbarkeit des Problems ist somit gefordert, dass die Anzahl der Mittel m gleich der Anzahl der Objekte n ist ($m = n$).

Eine typisches Anwendungsbeispiel ist die Zuordnung von Aufträgen zu Maschinen. Es gibt genau n Aufträge und n Maschinen. Jeder Maschine soll genau ein Auftrag erhalten. Die Produktionskosten seien bekannt und könnten sich zum Beispiel aus der Bearbeitungsdauer ableiten. Jede Maschine benötigt eine andere Zeit zur Bearbeitung eines gewissen Auftrages. Es ist nun die Zuordnung gesucht, bei der die Summe der Bearbeitungsdauern minimal ist.

Es sei x_{ij} die binäre Zuordnungsvariable:

$$x_{ij} = \begin{cases} 1 & \text{, wenn Mittel } i \text{ dem Objekt } j \text{ zugeordnet wird} \\ 0 & \text{, sonst} \end{cases}$$

Zudem sei $\forall i, j$ mit c_{ij} der Kosten- bzw. Nutzwert der Zuordnung gegeben. Es ergibt sich folgendes lineares Modell:

(Z)

ZF:
$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \rightarrow opt \tag{4.12a}$$

unter den NB:
$$\sum_{j=1}^{n} x_{ij} = 1 \qquad i = 1, \ldots, n \tag{4.12b}$$

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad j = 1, \ldots, n \tag{4.12c}$$

$$x_{ij} \in \{0, 1\} \qquad i, j = 1, \ldots, n \tag{4.12d}$$

Mit Hilfe der Nebenbedingung (4.12b) wird jedem Mittel genau ein Objekt zugeordnet. Analog wird durch die Nebenbedingung (4.12c) jedem Objekt genau ein Mittel zugeordnet.

Dies ist ein Problem aus der binären Optimierung und es existieren insgesamt $n!$ zulässige Lösungen. Zur Lösung des Zuordnungsproblems können die Verfahren zur Lösung des Transportproblems herangezogen werden. Da jedoch pro Zeile und Spalte in der Restriktionsmatrix nur eine Eins enthalten ist, handelt es sich um ein stark entartetes[5] Transportproblem. Die Lösung mit diesen Verfahren ist somit nicht sinnvoll.

Ein effizientes Verfahren zu Lösung des Zuordnungsproblems ist die Ungarische Methode. Diese Methode wendet die Matrixreduktion und eine Transformation auf die Kostenmatrix C so an, dass keine negative Elemente entstehen, aber genügend viele gleich Null sind. Im Abschnitt 4.4.3 wird der Algorithmus vollständig angegebene und ebenso die Matrizenreduktion erläutert.

Das lineare Zuordnungsproblem tritt in dieser speziellen Form selten in der Praxis auf. Eine Erweiterung des Modells zum verallgemeinerten Zuordnungsproblem ist die Zuweisung von mehreren Mitteln ($i = 1, \dots, m$) zu einem Objekt ($j = 1, \dots, n$) mit $m > n$, so werden zum Beispiel einem Arbeiter mehrere Aufträge zugeordnet. Jede Zuordnung von einem Mittel zu einem Objekt erhält ein Gewicht a_{ij} und zudem jedes Objekt eine Kapazitätsgrenze b_j. Somit ergibt sich das folgende Modell:

(vZ)

$$\text{ZF:} \qquad \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \to opt \tag{4.13a}$$

$$\text{unter den NB:} \qquad \sum_{j=1}^{n} x_{ij} = 1 \qquad i = 1, \dots, m \qquad m > n \tag{4.13b}$$

$$\sum_{i=1}^{m} a_{ij} x_{ij} \leq b_j \qquad j = 1, \dots, n \tag{4.13c}$$

$$x_{ij} \in \{0, 1\} \qquad i = 1, \dots, m, \; j = 1, \dots, n \tag{4.13d}$$

Die erste Nebenbedingung (4.13b) fordert, dass jedem Mittel genau ein Objekt zugeordnet wird und die NB (4.13c), dass jedem Objekt mehrere Mittel, aber unter Beachtung der Kapazitätsgrenze, zugeordnet werden können.

Das verallgemeinerte Zuordnungsproblem gehört zu der Klasse der NP[6]-Probleme und lässt sich nach derzeitigen Wissensstand nicht in polynomieller Zeit lösen. Das Aufsuchen der optimalen Lösung mit exakten Verfahren ist bei größeren Problemen in akzeptabler Zeit nicht möglich. Aus diesem Grund werden heuristische Näherungsverfahren bevorzugt, zum Beispiel das Eröffnungsverfahren: verbessertes Regret-Verfahren und des Verbesserungsverfahren: Tabu Search. Eine ausführliche Beschreibung der Anwendung dieser Verfahren auf das verallgemeinerten Zuordnungsproblem ist im Abschnitt 4.4 zu finden.

[5]Im Lösungsprozess können Zyklen entstehen, welche ein Erreichen der optimalen Lösung verhindert [9].

[6]Ein Problem liegt in der Klasse NP, wenn es ein polynomiellen Algorithmus gibt, der entscheidet ob eine gefundene *Lösung* tatsächlich eine Problemlösung ist. Es ist derzeit kein Algorithmus bekannt, der auch das schwierigste Problem diesen Typs in polynomieller Zeit löst.

4.3.3 Rundreiseprobleme

Das Rundreiseproblem, auch Problem des Handlungsreisenden oder Traveling Salesman Problem (kurz: TSP) genannt ist eines der am bekanntesten und meist untersuchten diskreten Optimierungsprobleme. Es beruht auf einem anschaulichen praktischen Problem:

Beispiel TSP Gegeben seien n Städte mit den Entfernungen c_{ij} zwischen den Städten i und j. Ein Handlungsreisender möchte alle diese Städte anfahren um seine Waren zu verkaufen und am Ende wieder an den Ausgangspunkt zurückkehren. Gesucht ist nun eine minimale Rundreise, d. h. die Reihenfolge der Besuche der Städte, so dass der zurückgelegte Weg minimal ist.

Ein Beispiel für eine Rundreise des Handlungsreisenden ist in der Abbildung 4.1 dargestellt. Dabei werden die sächsische Städte in der Reihenfolge Chemnitz, Mittweida, Freiberg, Marienberg, Stollberg, Crimmitschau, Limbach-Oberfrohna und wieder Chemnitz angefahren.

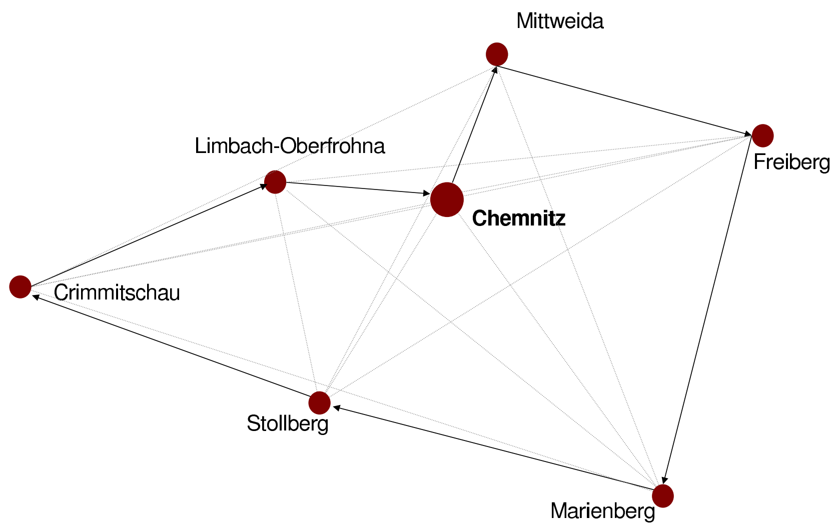


Abbildung 4.1: Rundreise durch durch 7 Städte

Obwohl dieses Problem zunächst sehr einfach erscheint, gehört es der Komplexitätsklasse NP an. Folgenden Zahlen verdeutlichen die Schwierigkeit der Lösung des Problems: beim Beispiel TSP existieren bei diesen 7 Städten $\frac{(7-1)!}{2} = 2520$ verschiedene Rundreisen, beim einfachen Rundreiseproblem[7] mit 10 Städten schon 181440 und bei 50 Städten sind es bereits $3,04 \cdot 10^{62}$. Eine Lösung mit Hilfe der vollständigen Enumeration erscheint somit nicht sinnvoll. Aufgrund dieses exponentiellen Anstiegs der Anzahl an Rundreisen wird bei praktischen Problemen meist auf exakte Verfahren verzichtet. Es existiert eine hohe Anzahl an Näherungs-, heuristischen und metaheuristische[8] Verfahren, so z. B. das Branch&Cut-Verfahren [11], die Lin-Kernighan-Heuristik [15] und die Lokale Suche [25].

[7] Es wird vorausgesetzt, dass zwischen zwei Städten immer eine Verbindung existiert, bei der die Fahrtrichtung keine Rolle spielt.

[8] Ein metaheuristisches Verfahren ist ein allgemeines Verfahren zur Ermittlung einer Näherungslösung eines kombinatorischen Optimierungsproblem, welches allgemein formuliert wird und auf ein konkretes Problem zugeschnitten werden muss.

Bevor auf ein mögliches Lösungsverfahren eingegangen wird, muss das TSP näher beschrieben und eine Modellierung angegeben werden. So kann das Problem zum Beispiel graphentheoretisch betrachtet werden. Gegeben sei ein zusammenhängender[9] Graph $G = \{V, E\}$ mit der Menge der Knoten $V = \{1, \ldots, n\}$ und der Menge der Kanten $E = \{(k,l)\}_{k,l \in V}$. Die Knoten spiegeln die Städte wider und die Kanten die Wege zwischen den Städten. Der Graph ist gewichtet, d. h. jede Kante erhält ein Gewicht, welche die Entfernungen c_{ij} sein können. Gesucht ist nun ein Hämiltonkreis, welcher eine minimaler Gesamtlänge besitzt. Ein Hämiltonkreis ist ein Kreis, der alle Knoten aus V enthält [10].

Das Rundreiseproblem wird in zwei Klassen unterschieden, dem symmetrischen und dem asymmetrischen TSP. Beim symmetrischen Rundreiseproblem ist der Graph G ungerichtet, d. h. für jede Kante $e = (i, j)$ gilt $c_{ij} = c_{j\,i}$. Im Gegenzug handelt es sich bei asymmetrischen TSP um einen gerichtete Graphen und es existieren Knoten $i, j \in V$, für die gilt $c_{ij} \neq c_{ji}$. Im Folgenden werden ausschließlich symmetrische TSP betrachtet.

Viele Algorithmen sind nur auf symmetrische $\Delta-$TSP anwendbar. Beim $\Delta-$TSP gilt die Dreiecksungleichung, welche besagt, dass für je drei Knoten $i, j, k \in V$ die Relation

$$c_{ij} \leq c_{ik} + c_{kj} \tag{4.14}$$

erfüllt ist. Auf das Anwendungsbeispiel bezogen bedeutet dies, dass die Kosten einer direkten Fahrt von i nach j höchstens genauso groß sind wie die Summe der Kosten, wenn von i nach j über den Ort k gefahren wird. Dabei gilt, dass jedes TSP mit der Kostenmatrix $C = (c_{ij})_{n \times n}$ sich durch Addition einer geeigneten Konstante $B \in \mathbb{R}$ in ein $\Delta-$TSP transformieren lässt [10]. Somit ist die Forderung, dass das gegebene Problem ein $\Delta-$TSP sein muss, keine Einschränkung.

Die Kostenmatrix $C = (c_{ij})_{n \times n}$ ergibt sich aus den Kosten, welche entstehen, wenn die Kante vom Knoten i nach j benutzt wird. Können zwei Städte nicht direkt untereinander erreicht werden, dann werden die Kosten

$$c_{ij} = \min_{w \in W} \{ \sum_{(k,l) \in w} c_{kl} \} \tag{4.15}$$

gesetzt, wobei W die Menge aller Wege zwischen den Städten i und j ist. Somit kann im Folgenden von einem vollständigen[10] Graph ausgegangen werden in dem nicht vorhandene Kanten hinzugefügt und mit den Kosten aus (4.15) gewichtet werden. Die Elemente c_{ii} der Hauptdiagonalen der Kostenmatrix werden auf Unendlich gesetzt, da eine Fahrt zur selben Stadt nicht zweckmäßig ist.

Eine Beschreibung der Rundreise kann über die Variable x_{ij} erfolgen:

$$x_{ij} = \begin{cases} 1 & \text{, falls die Kante zwischen dem Knoten } i \text{ und } j \text{ in der Rundreise enthalten ist} \\ 0 & \text{, sonst.} \end{cases} \tag{4.16}$$

Es lässt sich das folgende lineare binäre Optimierungsmodell aufstellen:

[9] Ein Graph $G = \{V, E\}$ ist zusammenhängend, wenn zwischen je zwei Knoten aus V ein Weg mit Kanten aus E existiert.

[10] In einem vollständigen Graph $G = \{V, E\}$ existiert zwischen allen Knoten aus V eine Kante $e \in E$. Die Anzahl der Kanten beträgt $\frac{|V||V-1|}{2}$.

(TSP)

$$\text{ZF:} \quad \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}\, x_{ij} \rightarrow min \tag{4.17a}$$

$$\text{unter den NB:} \quad \sum_{j=1}^{n} x_{ij} = 1 \qquad i = 1, \ldots, n \tag{4.17b}$$

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad j = 1, \ldots, n \tag{4.17c}$$

$$\sum_{i \in S}\sum_{j \notin S} x_{ij} \geq 1 \qquad \forall S \subset \{1, 2, .., n\} \tag{4.17d}$$

$$x_{ij} \in \{0, 1\} \qquad i = 1, .., n,\ j = 1, .., n \tag{4.17e}$$

Die Zielfunktion (4.17a) spiegelt die Minimierung der Gesamtkosten über alle zur Rundreise gehörenden Kanten wieder. Die Nebenbedingungen (4.17b) fordert, dass in der Rundreise jeder Knoten nur einmal Endknoten und (4.17c), dass jeder Knoten nur einmal Startknoten ist, d. h. bei der Rundreise führt genau eine Kante in den Knoten i hinein und eine Kante aus dem Knoten i heraus ($\forall i \in V$). Die dritte Nebenbedingung wird auch als *Kurzzyklennebenbedingung* bezeichnet. Mit Hilfe der Forderung (4.17d) werden Kurzzyklen verhindert. Kurzzyklen sind Kreise, bei denen jedoch noch nicht jeder Knoten besucht wurde. Sie haben eine Länge, die kleiner als $n - 1$ ist.

Die Anzahl der Kurzzyklennebenbedingung beträgt $2^{(n-1)} - n - 1$ und damit ist eine Auflistung aller Nebenbedingungen bereits sehr aufwendig. Aus diesem Grund werden in den meisten Lösungsverfahren nicht alle Nebenbedingungen aufgelistet, sondern während der Ausführung Kurzzyklentests durchgeführt, d. h. überprüft, ob in den Teillösungen Kurzzyklen aufgetreten sind.

Wie bereits erwähnt, ist die Anzahl der Lösungsverfahren immens, im Abschnitt 4.4.7 wird ein mögliches Verbesserungsverfahren, dass 2-opt-Verfahren als Verbesserungsverfahren für das TSP näher erläutert. Als Eröffnungsverfahren kann die *Greedy-Methode* angewendet werden. Dabei werden die Entfernungen aufsteigend sortiert. Es wird in jedem Schritt die Kante zur Lösung hinzugenommen, welche der Lösung noch nicht angehört, die kleinste Entfernung hat und bei deren Hinzunahme kein Kurzzyklus entsteht. Eine zulässige Startlösung für das Verbesserungsverfahren wurde erreicht, wenn ein Hämiltonkreis entstanden ist.

4.3.4 Tourenprobleme

Die Tourenprobleme, auch Vehicle Routing Problems (VRP) genannt, beschreiben eine ganze Klasse von Problemen und sind eine Kombination aus Zuordnungs- und Rundreiseproblemen. Im Rahmen dieser Arbeit wird vom folgenden Grundproblem ausgegangen [13].

Beispiel VRP Gegeben seien wieder n Städte und der Wohnort des Handlungsreisenden wie in Beispiel TSP. Nun ist die Anzahl der Städte n mit ihren Verweildauern p_i jedoch so groß, dass der Handlungsreisende sie nicht alle an einem Tag abarbeiten kann. Er benötigt mehrere Tage für den Besuch aller Städte, wobei er am Ende des Tages wieder zum Wohnort zurückkehrt. Gesucht ist die Zuordnung der Städte zu den Tagen und eine Reihenfolge der Besuche der Städte an einem Tag, so dass die Gesamtstrecke minimal ist und zudem die Arbeitszeit eines jeden Tages nicht überschritten wird. Eine *Tour* T_k ist die Rundreise des Tages k.

Neben der Verweildauer in den Städten muss die Fahrzeit zwischen den Städten mit berücksichtigt werden. Wird vorausgesetzt, dass die Fahrzeit linear abhängig zur Fahrstrecke ist, kann auch eine Minimierung der Gesamtfahrzeit anstelle der Gesamtfahrstrecke vorgenommen werden.

In Abbildung 4.2 sind 11 Städte mit Chemnitz als Wohnort des Handlungsreisenden gegeben. Alle Städte an einem Tag anzufahren ist für den Handlungsreisenden nicht möglich. Er entscheidet sich drei Tage einzuplanen. Er hat keinerlei Vorgaben wann oder in welche Reihenfolge er die Städte anfahren muss. Eine mögliche Tourenbildung ist in der Abbildung 4.2 dargestellt.

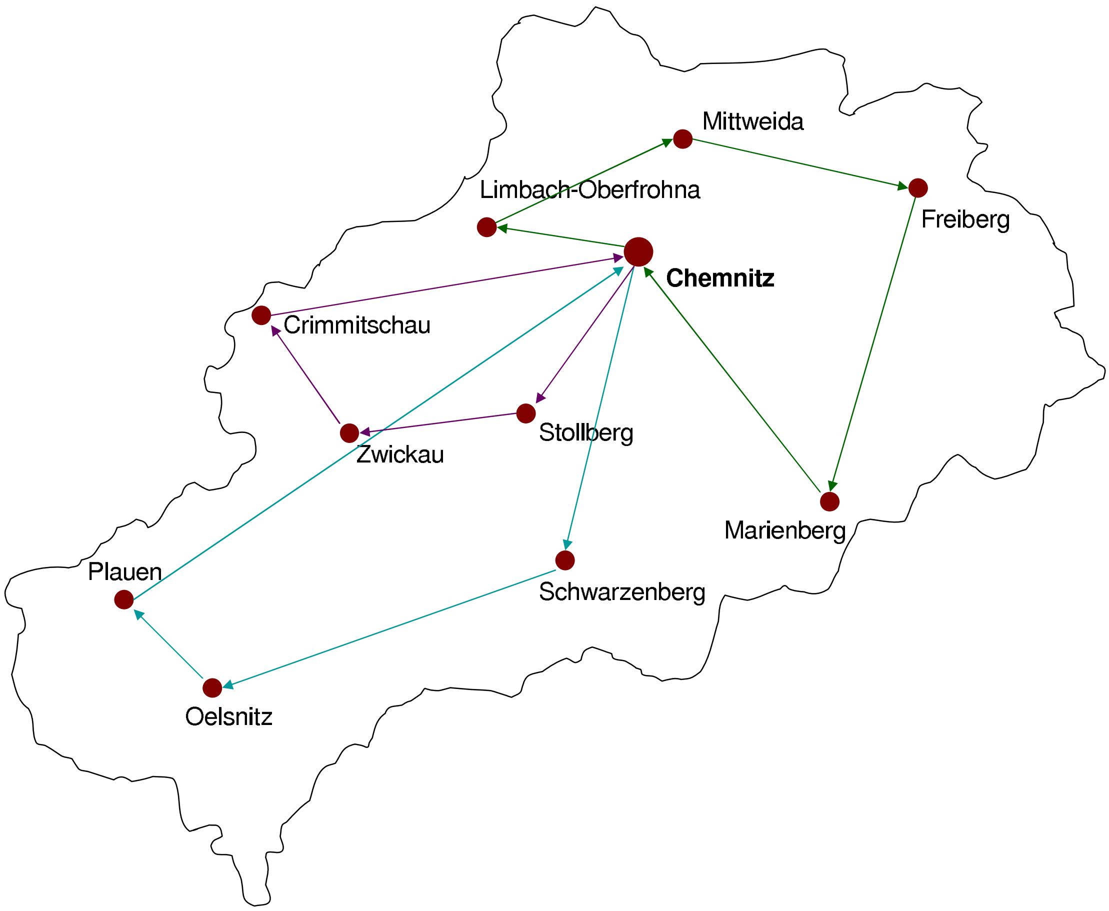


Abbildung 4.2: Mögliche Lösung für das Tourenproblem mit 10 Städten und dem Wohnort Chemnitz

Der Wohnort des Handlungsreisenden wird auch als Depot D bezeichnet. Das Beispiel VRP ist somit ein Eindepot-Tourenproblem. Die m verschiedenen Tage in denen der einzelnen Handlungsreisende fährt, können ebenso als m gleichwertige verschiedene Handlungsreisende betrachtet werden. Es sind m Touren gesucht, welche eine minimale Gesamtfahrzeit verzeichnen.

Die Vielzahl der Varianten des Problems ist bereits an diesem kleinen Beispiel zu erkennen. So könnte nicht die Arbeitszeit des Handlungsreisenden eingeschränkt sein, sondern die Ladekapazität eines Gutes und in jeder Stadt erwartet er eine gewisse Verkaufsmenge an diesem Gut. Diese

Art von Problemen zählen zu den *kapazitätsbeschränkten* Tourenproblemen (Capacitated vehicle routing problem (CVRP)).

Im Folgenden werden jedoch *längenbeschränkte* Tourenprobleme (Distance constraint vehicle routing problem (DVRP)) näher betrachtet, welchen auch das Beispiel VRP angehört.

Es sei

$N = \{0, 1, 2, \dots, n\}$ - die Menge aller Städte mit dem Depot 0

$A \in \mathbb{R}^+$ - die Arbeitszeitkapazität des Handlungsreisenden

$C = (c_{ij})_{i,j \in N}$ - die Matrix mit den Fahrzeiten zwischen den Städten i und j

$P = (p_i)_{i \in N \setminus \{0\}}$ - der Vektor mit den Verweildauern in den Städten

$T_k = \{k_1, \dots, k_{n_k}\}$ - eine zulässige Tour mit $k_i \in N \setminus \{0\}$.

Eine Tour ist in diesem Fall zulässig, wenn die Summe der Fahrzeiten B_k der Tour die Arbeitszeit A nicht überschreitet. Die Menge aller zulässigen Touren M lässt sich wie folgt beschreiben:

$$M = \{T_k \mid T_k \subseteq N \setminus \{0\} \wedge B_k \leq A,\ k = 1, \dots, m\} \tag{4.18}$$

$$\text{mit: } B_k = c_{0,k_1} + c_{k_{n_k},0} + \bigcup_{i=1}^{n_k - 1} c_{k_i k_{i+1}} \tag{4.19}$$

Bisher wurden bei den Touren jedoch die Verweildauer in den Städten vernachlässigt. Diese können in der Matrix C mit berücksichtigt werden, in dem jeweils die Hälfte der Verweildauern der Knoten i und j zur Fahrzeit c_{ij} addiert wird. Die neuen Einträge der Fahrzeitenmatrix ergeben sich somit zu:

$$c_{ij} := \frac{p_i}{2} + c_{ij} + \frac{p_j}{2} \tag{4.20}$$

Nicht in jedem Fall existiert eine zulässige Lösung für das Tourenproblem. Ist jedoch zusätzlich die Bedingung

$$c_{0k} + c_{k0} \leq A \qquad \forall k \in N \setminus \{0\} \tag{4.21}$$

erfüllt, so existiert eine solche. In Worten bedeutet diese Bedingung, dass es für alle Städte mögliche sein muss, an einem Tag ein Stadt anzufahren, abzuarbeiten und zurück zu fahren ohne die Arbeitszeitgrenze zu verletzten.

Für die Modellierung der DVRP werden zusätzlich die binären Variablen a_{ik} und x_k benötigt. Diese sind wie folgt definiert:

$$a_{ik} = \begin{cases} 1, & \text{wenn die Stadt } i \text{ zur Tour } T_k \text{ gehört} \\ 0, & \text{sonst} \end{cases} \tag{4.22}$$

$$x_k = \begin{cases} 1, & \text{wenn die Tour } T_k \text{ in der Lösung enthalten ist} \\ 0, & \text{sonst} \end{cases} \tag{4.23}$$

Mit Hilfe dieser beiden Variablen lässt sich nun das *Set-Covering-Modell* zum DVRP aufstellen:

$(DVRP_1)$

$$\text{ZF:} \quad \sum_{k:T_k \in M} B_k\, x_k \rightarrow min \tag{4.24a}$$

$$\text{unter den NB:} \quad \sum_{k:T_k \in M} a_{ik} x_k = 1 \qquad i = 1, \dots, n \tag{4.24b}$$

$$x_k \in \{0, 1\} \qquad \forall\, T_j \in M \tag{4.24c}$$

Die Zielfunktion minimiert die Gesamtfahrzeit über und innerhalb aller Touren. Da in der Menge M nur alle zulässigen Touren enthalten sind, wird nur noch die Bedingung (4.24b) benötigt. Diese gewährleistet, dass jede Stadt in genau einer Tour enthalten ist.

Das Set-Covering-Modell ist eine kompakte Formulierung des Tourenproblems. Jedoch darf nicht vernachlässigt werden, dass die Bildung der Menge M, also der Menge alle zulässigen Touren, bereits einen exponentiellen Aufwand besitzt. Neben dem Set-Covering-Modell existieren weitere Modellformen, so zum Beispiel das lineare binäre Modell zur Tourenplanung. Beim linearen Modell wird das Problem wie bereits beim Rundreiseproblem graphentheoretisch interpretiert. Dabei bildet die Menge der Städte die Knotenmenge und die Kosten c_{ij} aus Gleichung (4.20) sind die Kantengewichte im Graphen. Der Graph ist vollständig, da alle Städte untereinander erreichbar sind. Wird eine Fahrt zwischen zwei Städten i und j nicht erwünscht bzw. existieren weitere Restriktionen, die diese verbieten, wird $c_{ij} = M_\infty$ gesetzt.

Sei T_k die k. Tour der Lösung. Sie enthält außer dem Depot alle Städte und deren Reihenfolge, wie sie am k.ten Tag angefahren werden. Für die Aufstellung des linearen Modells werden die binären Variablen x_{ijk} und y_{ik} benötigt. Diese lassen sich wie folgt interpretieren:

$$x_{ijk} = \begin{cases} 1, & \text{wenn in der Tour } T_k \text{ die Kante } (i, j) \text{ befahren wird.} \\ 0, & \text{sonst} \end{cases} \tag{4.25}$$

$$y_{ik} = \begin{cases} 1, & \text{wenn der Knoten } i \text{ in der Tour } T_k \text{ enthalten oder } i = 0 \text{ ist.} \\ 0, & \text{sonst} \end{cases} \tag{4.26}$$

Um zu sichern, dass in jeder Tour das Depot enthalten ist, wird $y_{0k} = 1\, \forall k$ gesetzt. Für $i = 0$ ist y_{ik} somit keine Optimierungsvariable, sondern festgelegt.

Das lineare binäre Modell ergibt sich nun wie folgt:

$(DVRP_2)$

$$\text{ZF:} \quad \sum_{k=1}^{m}\sum_{i=0}^{n}\sum_{j=0}^{n} c_{ij}\, x_{ijk} \rightarrow min \tag{4.27a}$$

$$\text{unter den NB:} \quad \sum_{i=0}^{n}\sum_{j=0}^{n} c_{ij} x_{ijk} \leq A \qquad \forall k = 1,\dots,m \tag{4.27b}$$

$$\sum_{k=1}^{m} y_{ik} = 1 \qquad \forall i = 1,\dots,n \tag{4.27c}$$

$$\sum_{i=0}^{n} x_{ijk} \leq y_{ik} \qquad \forall k = 1,\dots,m;\ j = 0,\dots,n \tag{4.27d}$$

$$\sum_{j=0}^{n} x_{ijk} \leq y_{jk} \qquad \forall k = 1,\dots,m;\ i = 0,\dots,n \tag{4.27e}$$

$$\sum_{(i,j)\in S} x_{ijk} \leq |S| - 1 \qquad S \subset N; |S| \geq 1; k = 1,\dots,m \tag{4.27f}$$

$$x_{ijk} \in \{0,1\} \qquad \forall\ i,j = 0,\dots,n;\ k = 1,\dots,m \tag{4.27g}$$

$$y_{0k} = 1 \wedge y_{ik} \in \{0,1\} \qquad \forall\ i = 1,\dots,n;\ k = 1,\dots,m \tag{4.27h}$$

Die Zielfunktion minimiert die Summe der Kosten über alle in den einzelnen Touren verwendeten Kanten. Die Nebenbedingung (4.27b) fordert, dass für jede einzelne Tour die Arbeitszeitkapazität nicht überschritten wird. Die zweite Nebenbedingung gewährleistet, dass jeder Konten nur einmal in einer Tour enthalten ist. Eine Ausnahme bildet das Depot ($i = 0$), dieses wird in jeder Tour angefahren. Die beiden Nebenbedingungen (4.27d) und (4.27e) stellen sicher, dass zu jedem Knoten, welcher in einer Tour enthalten ist genau eine Kante in dieser Tour hinführt und wegführt. Die fünfte Nebenbedingung vermeidet analog zur Nebenbedingung (4.17d) die Kurzzyklen innerhalb einer Tour.

Es wurden hier zwei verschiedene Modelle für das längenbeschränkte Tourenproblem vorgestellt. Eine weiterführende Erläuterung und weitere Modelle sind unter anderem in der Literatur [15] und [13] angegeben.

Das Tourenproblem besteht aus zwei Teilproblemen:

- Zuordnungsproblem: Die Zuordnung der Städte zu den einzelnen Touren.
- Rundreiseproblem: Die Bestimmung der minimalen Rundreise zwischen den Städten für jede einzelne Tour.

Jedoch können diese beiden Probleme nicht getrennt voneinander betrachtet werden, sondern sie beeinflussen sich gegenseitig. Dies hat zur Folge, dass die Ermittlung der optimalen Lösung für praktische Probleme in vertretbarer Zeit nicht möglich ist, da das Rundreiseproblem der Komplexitätsklasse NP angehört. Aus diesem Grund werden bei praktischen Problemen, wie bereits beim Rundreiseproblem, heuristische Näherungsverfahren bevorzugt.

Wenn die oben beschriebenen zwei Teilprobleme nacheinander gelöst werden, sind dies sukzessive Verfahren. Diese werden wiederum in *Route-First-Cluster-Second* und *Cluster-First-Route-Second-Verfahren* unterschieden. Beim Route-First-Cluster-Second-Verfahren wird zunächst eine Rundreise über alle Knoten gebildet und im Anschluss diese „Giant-Tour" in kleinere zulässige Touren zerlegt. Beim Cluster-First-Route-Second-Verfahren wird zunächst eine Zuordnung der Knoten zu den Touren vorgenommen und im Anschluss die Reihenfolge der Knoten innerhalb einer Tour festgelegt. Beispiele für diese Verfahrenstypen und Hinweise, wann diese Verfahren geeignet sind, können unter anderem in [16] nachgeschlagen werden.

Ein weitere Möglichkeit für die Ermittlung einer zulässigen Startlösung sind die *parallelen* Verfahren. Dabei werden beide Teilprobleme gleichzeitig betrachtet. Ein Vertreter ist das *Savings-Verfahren*, in dessen Ergebnis eine zulässige Lösung für das DVRP erhalten wird. Der Algorithmus wird in 4.4.6 näher erläutert.

Es ist somit ein Eröffnungsverfahren, an das als Verbesserungsverfahren ein Lokale Suche angeschlossen werden kann. Hier findet eine Unterscheidung zwischen toureninterner und tourenübergreifende Verbesserung statt. Bei der toureninternen Verbesserung, werden ausschließlich Knoten innerhalb einer Tour betrachtet und nach verbesserten Lösungen gesucht. Bei der tourenübergreifenden Verbesserung findet ein Knotenaustausch zwischen den Touren statt. Das 2 bzw. 3-opt-Verfahren ist ein typisches Verbesserungsverfahren für das Tourenproblem. Im Abschnitt 4.4.7 wird auf das 2 und 3-opt-Verfahren eingegangen und am Beispiel der Tourenplanung näher erläutert.

Das bisher vorgestellte Tourenproblem bildet die Grundlage für verschieden Spezialfälle, so kann zum Beispiel der Handlungsreisende Termine in einigen Städten haben oder darf die Städte nicht in jeder beliebigen Reihenfolge anfahren. Erweiterungen der DVRP und die Möglichkeiten der Modellierung werden in [16] angegeben.

4.4 Algorithmen

Im letzten Abschnitt wurden einige Probleme der diskreten Optimierung näher erläutert. Im Folgenden wird beschrieben, mit Hilfe welcher Methoden oder Verfahren diese gelöst werden können. Zunächst werden die exakten Verfahren erläutert wie z. B. die Potentialmethode oder das Branch&Bound-Verfahren. Im Anschluss wird auf das verbesserte Regret-Verfahren und das Savings-Verfahren als Eröffnungsverfahren eingegangen. Am Ende werden Tabu Search und das 2 bzw. 3-opt-Verfahren als Verbesserungsverfahren näher beschrieben. Die Grundideen der exakten Verfahren wurde aus verschiedener Literatur, z. B. [12] und [25], entnommen.

4.4.1 Vogelsche Approximationsmethode

Die Vogelsche Approximationsmethode[11] ist ein Eröffnungsverfahren für das klassische Transportproblem, welches eine zulässige Startlösung ermittelt, die nahe am Optimum liegt oder bereits optimal ist. Dabei werden nicht nur die kostengünstigsten Transportverbindungen betrachtet, sondern auch der zusätzliche Transportaufwand bei Nichtnutzung mit berücksichtigt. Dazu wird die Differenz zwischen zweitgünstigen und günstigen Kosten einer Verbindung für jeden Auslieferungs-

[11] siehe [12]

bzw. Bedarfsort berechnet. Zu dem Ort, für den diese Differenz am größten ist, wird die günstigste Verbindung aufgenommen.

Gegeben sei das klassische Transportproblem aus Abschnitt 4.3.1 mit der Modellierung (4.10). Zunächst wird in der Kostenmatrix C für jede Zeile (entspricht den Bedarfsorten) und Spalte (entspricht den Auslieferungsorten) die Differenz zwischen dem günstigsten und zweit günstigen Element berechnet. Im Anschluss wird in der Zeile k bzw. Spalte l mit der größten Differenz das kleinste Kostenelement c_{kl} gewählt. Existieren mehrere Zeilen bzw. Spalten mit der größten Differenz, dann wähle diese, die das kleinste Kostenelement enthält. Die zugehörige Variable x_{kl} wird gleich dem Minimum aus b_k und a_l gesetzt, d. h. vom Ort A_l wird zum Ort B_k das Minimum zwischen der Auslieferungsmenge a_l und der Bedarfsmenge b_k transportiert.

Wird $x_{kl} = a_l$ gesetzt, so bedeutet dies, dass die gesamte Auslieferungsmenge a_l zum Bedarfsort B_k transportiert wird. Der Auslieferungsort A_l verfügt über keine Ressourcen mehr. Somit kann die Spalte l gestrichen werden, d. h. $x_{rl} = 0\ \forall r \neq k$ und zudem muss die Bedarfsmenge $b_k = b_k - a_l$ angepasst werden. Wird im Gegenzug $x_{kl} = b_k$ gesetzt, dann ist der Bedarf von B_k gedeckt, die Zeile k kann gestrichen. Für die Variablen dieser Zeile gilt $x_{ks} = 0\ \forall s \neq l$ und die Auslieferungsmenge wird zu $a_l = a_l - b_k$ angepasst.

Eine zulässige Lösung ist erreicht, wenn alle Auslieferungsmengen leer sind bzw. alle Bedarfsmengen gedeckt sind, d. h. es gilt $\forall\, i\ :\ a_i = 0$ bzw. $\forall\, j\ :\ b_j = 0$. Vorausgesetzt wird, dass eine Lösung existiert. Dies kann mit Hilfe der Bedingung (4.9), dass die Summe der Auslieferungsmengen gleich die Summe der Bedarfsmengen ist, am Anfang getestet werden.

In vielen Fällen liefert die Vogelsche Approximation bereits eine optimale Lösung. Jedoch erkennt das Verfahren dies nicht. Aus diesem Grund wird die Potentialmethode an das Verfahren angeschlossen, um entweder die Optimalität der zulässigen Startlösung festzustellen oder mit deren Hilfe die optimale Lösung zu bestimmen.

Vogelsche Approximationsmethode

Schritt 0: Teste, ob die Bedingung (4.9) erfüllt ist.

- ja Gehe zur Initialisierung.
- nein Abbruch, das Transportproblem ist nicht lösbar. Ausgabe: $X = \emptyset$

Initialisierung: $X = (x_{ij})_{m,n}$ mit $x_{ij} = 0\ \forall i, j$

Schritt 1: Berechne die Differenz Δ_i zwischen dem zweitkleinsten und kleinsten Element für jede Zeile und jede Spalte.

Schritt 2: Wähle die Zeile bzw. Spalte mit der größten Differenz $\Delta_k = \max\{\max\{\Delta_i\}, \max\{\Delta_j\}\}$. Es sei in diesem Fall k eine Zeile, analog erfolgt der weitere Algorithmus, falls Δ_k in einer Spalte liegt.

Schritt 3: Ermittle das kleinste Element c_{kl} dieser Zeile.

Schritt 4: Gilt $a_l < b_k$?

- nein Setze $x_{kl} = b_k$, $x_{kr} = 0\ \forall r \neq l$, $a_l = a_l - b_k$, $b_k = 0$. Streiche Zeile k aus C.
- ja Setze $x_{kl} = a_l$, $x_{sl} = 0\ \forall s \neq k$, $b_k = b_k - a_l$, $a_l = 0$. Streiche Spalte l aus C.

Schritt 5: $\exists i : a_i > 0$?

nein Eine zulässige Lösung wurde gefunden. Ausgabe: X

ja Gehe zu Schritt 1.

Der Algorithmus der Vogelschen Approximation soll an einem kleinen Beispiel verdeutlicht werden. Gegeben seien drei Lager A_1, A_2, A_3 und vier Bedarfsort B_1, B_2, B_3, B_4 mit der Kostenmatrix C, dem Angebotsvektor a und Bedarfsvektor b:

$$C = \begin{pmatrix} 10 & 15 & 13 \\ 20 & 5 & 10 \\ 11 & 10 & 3 \\ 9 & 12 & 20 \end{pmatrix} \quad a = \begin{pmatrix} 100 \\ 150 \\ 120 \end{pmatrix} \quad b = \begin{pmatrix} 60 \\ 110 \\ 160 \\ 40 \end{pmatrix} \tag{4.28}$$

Die Bedingung (4.9) ist erfüllt, da sowohl die Summe des Bedarfs, also auch die Summe des Angebots gleich 370 beträgt. Im ersten Schritt wird die Differenz zwischen zweitkleinsten und kleinsten Element der Matrix C für jede Zeile und Spalte bestimmt. Dies sind die Werte in der Zeile Δ_i bzw. Spalte Δ_j. Zunächst besitzt die Zuordnungstabelle noch keine Einträge. Die Elemente, welche sich in der oberen Ecke befinden, spiegeln die Kosten der Zuordnung wider.

	A_1	A_2	A_3	b_i	Δ_i
B_1	^{10}	^{15}	^{13}	60	3
B_2	^{20}	^{5}	^{9}	110	4
B_3	^{11}	^{10}	^{3}	160	$\boxed{7}$
B_4	^{9}	^{12}	^{20}	40	3
a_j	100	150	120		
Δ_j	1	5	6		

Die größte Differenz zwischen dem zweitkleinsten und kleinsten Element beträgt 7 und wurde in der Tabelle markiert. Im nächsten Schritt wird der Zeile $i = 3$ der kleinste Eintrag gesucht. Dies ist $c_{32} = 3$. Der Wert der Zuordnung ergibt sich aus dem Minimum zwischen $b_3 = 160$ und $a_3 = 120$. Die Variable x_{33} wird 120 gesetzt und alle weitere Elemente der Spalte gleich Null, d. h. $x_{31} = x_{32} = x_{34} = 0$, da $a_3 < b_3$ gilt. Zudem wird $a_3 = 0$ bzw. $b_3 = 40$ angepasst. Die Δ_i bzw. Δ_j müssen neu berechnet werden. Es ergibt sich die folgende Tabelle:

	A_1	A_2	A_3	b_i	Δ_i	
B_1	^{10}	^{15}	0 ^{13}	60	3	5
B_2	^{20}	^{5}	0 ^{9}	110	4	15
B_3	^{11}	^{10}	120 ^{3}	40	7	1
B_4	^{9}	^{12}	0 ^{20}	40	3	3
a_j	100	150	–			
Δ_j	1	5	6			
	1	5	0			

Nun wird wieder das größte Δ_i bzw. Δ_j gesucht. In diesem Fall ist dies die Zeile $j = 2$ mit $\Delta_2 = 15$. Der Algorithmus wird analog fortgeführt, bis alle $a_j = 0$ sind. Das Ergebnis der Vogelschen Approximation ist in der folgenden Tabelle gegeben.

	A_1	A_2	A_3	b_i	Δ_i				
B_1	60^{10}	0^{15}	0^{13}	0	3	5	5	$-$	$-$
B_2	0^{20}	110^{5}	0^{9}	0	4	15	$-$	$-$	$-$
B_3	0^{11}	40^{10}	120^{3}	0	7	1	1	1	10
B_4	40^{9}	0^{12}	0^{20}	0	3	3	3	3	$-$
a_j	0	0	0						
Δ_j	1	5	6						
	1	5	$-$						
	1	2	$-$						
	2	2	$-$						
	$-$	10	$-$						

Die Startlösung für das Transportproblem ist somit $x_{11} = 60, x_{22} = 110, x_{32} = 40, x_{33} = 120, x_{41} = 40$. Die Kosten für diesen Transport betragen 2270.

4.4.2 Potentialmethode

Die Potentialmethode ermittelt iterativ für eine gegebene zulässige Startlösung für ein Transportproblem nach endlich vielen Schritten die optimale Lösung. Da für die Erläuterungen des Verfahrens die Dualitätstheorie und weitere Grundlagen der linearen Optimierung benötigt wurden, wird an dieser Stelle darauf verzichtet eine detaillierte Beschreibung zu geben und nur der Algorithmus angegeben. Für interessierte Leser empfiehlt sich die Literatur [12].

Für den Algorithmus wird das duale Problem von (4.10) benötigt. Dies lautet wie folgt:

(DTP)

ZF: $$\sum_{i=1}^{m} a_i u_i + \sum_{j=1}^{n} b_j v_j \rightarrow max \tag{4.29a}$$

unter den NB: $$u_i + v_j \leq c_{ij} \qquad i = 1, \ldots, m, j = 1, \ldots, n \tag{4.29b}$$

$$u_i, v_j \in \mathbb{R} \qquad i = 1, \ldots, m, j = 1, \ldots, n \tag{4.29c}$$

Potentialmethode

Schritt 1: Ermittle eine zulässige Anfangsbasislösung X z. B. mit Hilfe der Vogelschen Approximation.

Schritt 2: Setze $v_1 = 0$ und löse das Gleichungssystem $u_i + v_j = c_{ij}\ \forall i, j$ mit $x_{ij} > 0$ (Basisvariable)

Schritt 3: Test auf Optimalität: Gilt $g_{ij} = c_{ij} - u_i - v_j \geq 0\ \forall i, j$?

ja Die zulässige Basislösung ist optimal. Ausgabe: X

nein Gehe zu Schritt 4.

Schritt 4: Bestimme die Indices (k, l) mit $g_{kl} = \min_{i,j}\{g_{ij}\}$ und x_{kl} wird neue Basisvariable.

Schritt 5: Setze $x_{kl} = P$.

Schritt 6: Bestimme einen Pfad im Transporttablau aus Basisvariablen so, dass gilt

- der Pfad führt abwechselnd durch Zeilen und Spalte
- bildet bei Hinzunahme von x_{kl} einen Kreis
- Wert von der x_{ij} ändert sich abwechselnd um $-P$ und $+P$

Schritt 7: Setze $P = min\{x_{ij} : x_{ij}$ ist ein ungerades Element im Pfad$\}$. Ermittle das neue Transporttablau entlang des Pfades. Gehe zu Schritt 2.

4.4.3 Ungarische Methode

Die Ungarische Methode ist ein effizientes Verfahren zu Ermittlung der optimalen Lösung eines Zuordnungsproblems. Gegeben sei das Zuordnungsproblem (Z) gemäß (4.12) mit n Mitteln und n Objekten. Die Ungarische Methode beinhaltet die *Matrizenreduktion* und eine spezielle *Matrizentransformation*, welche die Kostenmatrix C so umformt, dass kein Matrixelement negativ, aber genügend Elementen gleich Null sind. Die optimale Lösung wurde gefunden, wenn n Element der Matrix, welche gleich Null sind, unabhängig sind.

Zunächst wird die Matrizenreduktions erläutert und im Anschluss auf die spezielle Matrixtransformation der Ungarischen Methode eingegangen. Die Darlegung des Algorithmus ist in der Literatur vielfältig. Die Erläuterung innerhalb dieser Arbeit wird an [20] angelehnt.

Matrizenreduktion

Die Matrixreduktion ist eine äquivalent Umformung der Kostenmatrix hinsichtlich des Optimierungsproblems, d. h. die optimale Lösung des Optimierungsproblems mit der Kostenmatrix C ist gleich der optimalen Lösung des Optimierungsproblems mit umgeformter Matrix $\widetilde{C}$.

Bei der Matrixreduktion wird in Gesamtreduktion, zeilenweise und spaltenweise Reduktion unterschieden. Bei der Gesamtreduktion wird das minimale Element c_{min} der gesamten Kostenmatrix ermittelt und die modifizierte Kostenmatrix $\widetilde{C}$ ergibt sich aus den Elementen $\widetilde{c}_{ij} = c_{ij} - c_{min}$ $\forall i, j$. Analog wird bei der zeilenweisen (spaltenweise) Reduktion das jeweilige Minimum der Zeile (Spalte) ermittelt und die Einträge von $\widetilde{C}$ ergeben sich zu:

$$\widetilde{c}_{ij} = c_{ij} - \min_j c_{ij} \quad \forall i \qquad \text{bzw.} \qquad \widetilde{c}_{ij} = c_{ij} - \min_i c_{ij} \quad \forall j \tag{4.30}$$

Im Allgemeinen ist die Gesamtreduktion nicht so effektiv wie das nacheinander Ausführen der spalten- und zeilenweisen Reduktion. Die Reihenfolge ist dabei nicht entscheidend, liefert jedoch im Allgemeinen unterschiedliche Matrizen $\widetilde{C}$. Nach Ausführung der beiden Reduktionen entsteht eine Matrix, welche mindestens eine Null pro Zeile und Spalte besitzt. Dies ist Grundlage für die spezielle Matrixtransformation.

Matrizentransformation der Ungarischen Methode

Das Ziel der speziellen Matrizentransformation ist es, eine Matrix zu erzeugen, die n unabhängige 0-Elemente besitzt, also Elemente der Matrix, welche gleich Null sind. Zunächst werden in der reduzierten Matrix $\widetilde{C}$ die maximale Anzahl an unabhängigen 0-Elementen gesucht. Dazu wird eine Spalte bzw. Zeile gewählt, welche die minimale Anzahl an 0-Elementen besitzt. Im nächsten Schritt wird eine Null aus dieser Spalte bzw. Zeile markiert und alle weiteren in dieser Zeile und Spalte werden gestrichen. Dies wird nun fortgeführt bis kein 0-Element in der Matrix mehr vorhanden ist. Die Anzahl der markierten 0-Elementen ist die maximale Anzahl an unabhängigen 0-Elementen.

Im Anschluss wird das *minimale Überdeckungssystem* gesucht, das ist ein System von Linien die alle Nullen überdecken und deren Anzahl minimal ist. Dazu werden folgende Schritte nacheinander durchgeführt

1 Markierung aller Zeilen, welche keine markierte Null enthalten

2 Markierung aller Spalten, welche eine gelöschte Null in einer bereits markierten Zeile enthalten

3 Markierung der aller Zeilen, welche eine markierte Null in einer markierten Spalte enthalten.

Der Schritt 2 und 3 werden solange wiederholt, bis keine Markierung von Spalten oder Zeilen mehr möglich ist. Danach werden alle Zeilen, welche *nicht* markiert wurden und alle Spalten, welche markiert wurden gestrichen. Das minimale Überdeckungssystem wurde gefunden.

Nun wird das kleinste Element der verbleibenden Matrix gesucht. Dieses muss größer Null sein, da bereits alle Nullen gestrichen wurden. Sei c_{min} dieses minimale Element. Von den verbleibenden Elementen der Matrix wird c_{min} subtrahiert und zu den Elementen, welche sowohl zeilenweise als auch spaltenweise gestrichen wurden, wird $2 \cdot c_{min}$ addiert.

Die Transformation wird nun wieder von vorne begonnen, in dem alle Markierungen, Löschungen und Streichungen aufgehoben werden und die maximale Anzahl an 0-Elementen bestimmt wird. Dies wird solange durchgeführt, bis die maximale Anzahl an 0-Elementen gleich der Anzahl der Zeilen n ist. Die optimale Lösung ist lässt sich aus diesen unabhängigen 0-Elementen ablesen.

Die Ungarische Methode ist sehr anschaulich und lässt sich per Hand gut nachvollziehen. Jedoch ist die Umsetzung in einem Programm schwieriger. Die Komplexität der Ungarischen Methode beträgt $O(n^4)$.

Damit ergibt sich die Zusammenfassung des Algorithmus für das Zuordnungsproblem (Z) aus (4.12) zu:

Ungarische Methode

Schritt 1: Durchführung der zeilenweise Reduktion.

Schritt 2: Durchführung der spaltenweise Reduktion.

Schritt 3: Bestimmung der maximal unabhängigen Anzahl an 0-Elementen.

Schritt 4: Entspricht die maximale Anzahl der unabhängigen 0-Elementen gleich n?

nein Markiere diese 0-Elementen.

ja Die optimale Lösung ist gefunden. Die unabhängigen 0-Elementen entsprechen der Zuordnung.

Schritt 5: Bestimme das minimale Überdeckunssystem durch streichen der entsprechenden Zeilen und Spalten (wie oben beschrieben).

Schritt 6: Bestimme das kleinste Element c_{min}, der nicht gestrichenen Element der Matrix C.

Schritt 7: Subtrahiere c_{min} von allen nicht gestrichen Elementen. Addiere $2 \cdot c_{min}$ auf Elemente, welche sowohl zeilenweise als auch spaltenweise gestrichen wurden.

Schritt 8: Hebe die Markierungen, Löschungen und Streichungen auf und gehe zu Schritt 3.

4.4.4 Branch&Bound-Methode

Die Branch&Bound-Methode ist eine exaktes Methode zur Lösung von diskreten Optimierungsaufgabe und kann unter anderem in [6] nachgeschlagen werden. Zur Erläuterung der Methode wird eine binäre Minimierungsaufgabe herausgegriffen. Die Grundidee der Branch&Bound-Methode besteht in der Zerlegung des Lösungsraumes in disjunkte Teilmengen. Jede Teilmenge wird darauf getestet, ob die optimale Lösung innerhalb dieser Teilmenge liegt. Ausschlusskriterien sind die *Unzweckmäßigkeit* und *Unzulässigkeit*. Eine Teilmenge ist unzulässig, wenn sie eine der Nebenbedingung verletzt. Eine Teilmenge ist unzweckmäßig, wenn die untere Schranke für den Zielfunktionswert des Teilproblems größer als ein für den optimalen Zielfunktionswert ermittelten oberen Schranke ist. Solche Mengen brauchen im Folgenden nicht mehr betrachtet werden, weil sie die optimale Lösung nicht enthalten können. Somit besteht das Ziel in der Zerlegung des Lösungsraumes in immer kleinere Mengen, welche nur weiter verzweigt werden, wenn kein Ausschlusskriterium zutrifft.

Die Zerlegung in Teilmengen wird als Verzweigung bzw. *Branching* und der Test auf Zweckmäßigkeit wird als *Bounding* bezeichnet. Damit ist die Branch&Bound-Methode eine unvollständige Enumeration, welche im Wesentlichen aus zwei Aufgaben dem Branching und dem Bounding besteht.

Beschreibung des Verfahrens

Betrachtet wird das folgende diskrete, lineare Minimierungsproblem:

(M)

$$\begin{aligned} &\text{ZF:} && \boldsymbol{c}^T\boldsymbol{x} \rightarrow min && && (4.31a)\\ &\text{unter den NB:} && \boldsymbol{x} \in R_D && && (4.31b)\\ & && x_i \in \{0,1\} && \forall i && (4.31c) \end{aligned}$$

Im ersten Schritt des Verfahren wird eine obere Schrank S_j mit $j = 0$ ermittelt. Dies kann auf verschiedene Arten und Weisen geschehen und ist vom Problem abhängig. Eine häufig verwendete obere Schranke ist der Zielfunktionswert einer zulässigen Lösung. Diese kann zum Beispiel durch Eröffnungsverfahren ermittelt werden. Im Allgemeinen gilt, je besser eine solche zulässige

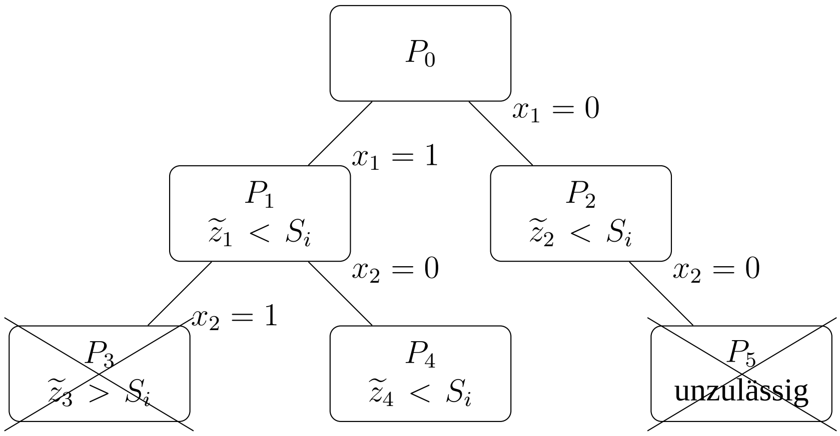


Abbildung 4.3: Darstellung der ersten Schritte des Branch&Bound-Verfahren als binären Baum

Anfangslösung und damit auch die obere Schranke ist, desto schneller wird beim Verfahren die optimale Lösung erreicht. Die obere Schranke wird während des Verfahrens verbessert.

Im zweiten Schritt wird der Restriktionsbereich R_D in zwei disjunkte Teilmengen R_1 und R_2 nach einer Verzweigungsregel zerlegt. Da das gegebene Problem binär ist, kann eine Variable x_k herausgegriffen werden und die Teilmengen R_1 wird mit $x_k = 1$ und R_2 mit $x_k = 0$ angenommen werden. Die Wahl der Variable x_k kann zufällig geschehen oder nach einer Heuristik. Die Verzweigung geschieht dem zufolge nach einer Regel, welche bei jedem Problem zunächst definiert werden muss.

Jede Teilmenge wird nun auf Zulässigkeit und Zweckmäßigkeit geprüft. Verstößt eine Teilmenge gegen eine der Nebenbedingungen wird sie wegen Unzulässigkeit gestrichen. Ist dies nicht der Fall wird ein relaxiertes Problem $\widetilde{R_i}$ betrachtet, welches einfach zu lösen ist, zum Beispiel das dazugehörige lineare Problem. Ist der Zielfunktionswert der optimalen Lösung von $\widetilde{R_i}$ größer als die obere Schrank S_j, wird dieses Teilproblem wegen Unzweckmäßigkeit ebenso gestrichen. Trifft keine der beiden Ausschlusskriterien zu, ist die Teilmenge R_i Ausgangspunkt für weitere Verzweigungen.

Veranschaulicht werden kann das Verfahren in einem binären Baum[12]. Die Knoten bilden die einzelnen Teilmengen. Von der *Wurzel* P_0 ausgehend verzweigen sich immer zwei weitere Knoten. Der Baum endet in einem Endknoten, welcher auch mit *Blatt* bezeichnet wird. Ein Blatt besteht aus einer einelementigen Teilmenge. Wird der Endknoten nicht wegen Unzulässigkeit gestrichen, dann impliziert dieser eine zulässige Lösung. Ein Beispiel für solch einen binären Baum ist in der Abbildung 4.3 dargestellt.

Ist der Zielfunktionswert eines Blattes kleiner als die obere Schranke S_i, wird dieser als neue obere Schranke S_{i+1} verwendet. Das Ende des Verfahrens ist erreicht, wenn keine Teilmenge mehr zu untersuchen ist. Die optimale Lösung enthält der Endknoten mit dem kleinsten Zielfunktionswert. Existiert kein zulässiger Endknoten, dann besitzt das Problem keine zulässige Lösung. Haben mehre Endknoten den gleichen minimalen Zielfunktionswert, so existieren mehrere optimale Lösungen.

Die Reihenfolge der Verzweigen kann nach verschiedenen Regeln erfolgen. Grundsätzlich wird in zwei Varianten unterschieden:

[12] Ein Baum ist ein zusammenhängender, kreisfreier Graph [10]. Ein binärer Baum besitzt eine Wurzel (Startknoten) mit zwei Nachfolgern. Jeder weitere Knoten besitzt einen Vorgänger und höchstens zwei Nachfolgern.

- Last-In-First-Out-Regel (LIFO-Regel)
- Minimum-Lower-Bound-Regel (MLB-Regel)

Bei der LIFO-Regel wird die zuletzt erhaltende Teilmenge zuerst verzweigt. Es wird immer nur eine Teilmenge untersucht und dann wieder verzweigt. Diese Vorgehen wird auch Tiefensuche genannt und liefert schnell eine zulässige Lösung und damit eventuell eine besser obere Schranke. Jedoch ist die Lösung meist weit vom Optimum entfernt. Dieses Vorgehen ist vor allem dann geeignet, wenn keine obere Schranke am Anfang ermittelt werden kann oder diese sehr grob ist.

Bei der MLB-Regel wird die Teilmenge weiter untersucht, welche die kleinste untere Schranke aufweist. Dieses Vorgehen ist heuristisch und die Hoffnung besteht darin, dass die optimale Lösung in der gewählten Teilmenge enthalten ist. Die Anwendung der MLB-Regel wird auch als Breitensuche bezeichnet. Eine erste Lösung wird nicht so schnell erhalten wie bei der LIFO-Regel, jedoch ist die erste zulässige Lösung im Allgemeinen näher am Optimum. Der Nachteil dieses Verfahren ist, dass der Speicheraufwand schnell wächst und auch die Umsetzung in einem Programm schwieriger als bei der LIFO-Regel ist.

Im Folgenden wird der Algorithmus des Branch&Bound-Verfahrens für binäre, lineare Minimierungsprobleme (M) angegeben.

Algorithmus B&B:

Schritt 1: Setze i=0 und k=1. Bestimme, wenn möglich, eine zulässige Lösung von (M) mit dem ZFW z_0. Setze die obere Schranke $S_i = z_0$ bzw. $S_i = \infty$. P_k sei die Wurzel des Lösungsbaums mit $k = 0$.

Schritt 2: Gibt es noch Knoten P_l, die noch nicht gestrichen wurden?

- nein Zulässige Lösung X mit minimalen Zielfunktionswert ist optimale Lösung. Ist keine Endknoten vorhanden, dann existiert keine zulässige Lösung. Ausgabe: X
- ja Wähle ein Knoten P_l nach der Reihenfolgeregelung und ein j mit $0 < x_j < 1, x_j \in X$. Verzweige den Knoten P_l. Es ergeben sich zwei neue Knoten P_k mit $x_j = 1$ und P_{k+1} mit $x_j = 0$. Setze $l = k$ und $k = k + 2$.

Schritt 3: Werden alle Nebenbedingungen von (M) im Knoten P_l bzw. P_{l+1} erfüllt?

- nein Streiche den Knoten P_l bzw. P_{l+1} der die NB nicht erfüllt wegen Unzulässigkeit.
- ja Gehe zu Schritt 4

Schritt 4: Berechne Zielfunktionswert $\widetilde{z}_l$ vom relaxierten Problem P_l bzw. P_{l+1}, zum Beispiel durch Weglassen der Ganzzahligkeitsforderung und Lösung des zugehörigen linearen Problems mittels Simplex-Algorithmus.

Schritt 5: Gilt $\widetilde{z}_l < S_i$ bzw. $\widetilde{z}_{l+1} < S_i$?

- nein Streiche Knoten P_l bzw. P_{l+1} wegen Unzweckmäßigkeit. Wenn beide Knoten gestrichen wurden, dann gehe zu Schritt 2 ansonsten zu Schritt 6.
- ja Gehe zu Schritt 6.

Schritt 6: Wurde ein Endknoten erreicht, d. h. $\nexists j$ mit $0 < x_j < 1$ in P_l bzw. P_{l+1}?

- nein Gehe zu Schritt 2.
- ja Berechne Zielfunktionswert z_l und z_{l+1}. Gilt $z_l < S_i$, dann setzte $i = i+1$ und $S_i = z_l$. Gilt $z_{l+1} < S_i$, dann setzte $i = i + 1$ und $S_i = z_{l+1}$. Gehe zu Schritt 2.

4.4.5 Verbessertes Regret-Verfahren

Für die Lösung des verallgemeinerten Zuordnungsproblems (vZ), bei dem unter Einhaltung von Kapazitätsgrenzen jedem Objekt mehrere Mittel zugeordnet werden (siehe Abschnitt 4.3.2), sind in der Literatur oft nur Verbesserungsverfahren angegeben, kaum jedoch Eröffnungsverfahren. Eine Methode zur Ermittlung einer Startlösung ist die Regret-Methode (vgl. [17] oder [12]). Das Verfahren lehnt sich an die Vogelsche Approximation (siehe Abschnitt 4.4.1) an, d. h. es wird iterativ zunächst das Mittel i gewählt, welches die größte Differenz in den Kosten c_{ij} zwischen zweitgünstigsten und günstigsten Objekt besitzt. Diese Differenz wird als *Regret* (engl.: bedauern) bezeichnet. Würde nun immer das günstigste Objekte dem Mittel i zugeordnet werden, so ist es möglich, dass das Verfahren in keiner zulässigen Lösung endet. Die Ursache ist die Nebenbedingung (4.13c), da die Kapazitäten der Objekte beschränkt sind. Aus diesem Grund wird bei der in [12] vorgestellten Regret-Methode zwischen dem Objekt k mit kleinstem c_{ik} oder Objekt l mit zweitkleinstem c_{il} zufällig gewählt und dem Mittel i zugeordnet. Genau dieser Zufall kann jedoch zu einer gerade ungünstige Zuordnung führen, da bei Wahl des Objektes mit dem zweitkleinsten c_{il} der Regret gerade am größten ist. Zudem garantiert der Zufall auch beim mehrmaligen Durchführen nicht, dass eine zulässige Startlösung gefunden wird.

Damit garantiert wird, das ein zulässige Lösung gefunden wird und diese auch nahe am Optimum wird hier eine Verbesserung der aus der Literatur bekannten Regret-Methode vorgestellt. Die Idee besteht darin, die Ermittlung der Lösung in einem binären Lösungsbaum, ähnlich wie bei der Branch&Bound-Methode, darzustellen. Die Branch&Bound-Methode (siehe Abschnitt 4.4.4) lässt sich in die Aufgaben des Verzweigens (Branching) und des Begrenzens (Boundings) unterteilen. Im verbesserten Regret-Verfahren wird nur die Idee des Verzweigens aufgegriffen und als Verzweigungsstrategie die Regret-Heuristik angewendet. Die Berechnung der Schranken und damit der Test auf Unzweckmäßigkeit wird nicht vorgenommen. Das Ziel des Verfahrens ist nicht die optimale Lösung zu finden, sondern eine gute zulässige Startlösung.

Ausgehend von einem Knoten wird so verzweigt, dass ein, nach der Regret-Heuristik gewähltes Mittel i einem Objekt j zu geordnet wird $(x_{ij} = 1)$ oder nicht $(x_{ij} = 0)$. Um eine zulässige Lösung möglichst schnell zu finden, wird die Tiefensuche als Reihenfolgeregel verwendet. Zur Ermittlung des Regrets $RG(i)$ werden zunächst die Kostenelemente c_{ij} eines Mittels i aufsteigend sortiert und $I_i = \{j_1, j_2, \ldots, j_m\}$ sei die zugehörige Indexmenge. Der Wert c_{ij_h} ist somit das h. kleinste Kostenelement eines Mittels i. $RG(i)$ lässt sich über die Differenz des zweitgünstigen und günstigen Kostenelements für jedes Mittel berechnen:

$$RG(i) = c_{ij_2} - c_{ij_1} \tag{4.32}$$

Zum Verzweigen wird diejenige Variable $x_{kl} = 1$ gesetzt, für die k das Mittel mit dem größten Regret ist und l das Objekt, welches in der Spalte k das kleinste Kostenelement besitzt. Neben der Zuordnung des Mittels k zum Objekt l, muss die Kapazitätsgrenze b_l des Objektes l angepasst werden. Der Parameter berechnet sich neu zu $b_l^{t+1} := b_l^t - a_{kl}$, wobei t dem t-ten Knoten des Lösungsbaums entspricht. Zudem wird der Regret $RG(k) = -1$ gesetzt, damit wird sicher gestellt, dass im Teilbaum das Mittel k nicht noch einmal zum Verzweigen gewählt wird.

Gilt in einem Knoten $a_{kl} > b_l^t$, kann der Knoten t wegen Unzulässigkeit gestrichen werden, da die Nebenbedingung (4.13c) verletzt wird. Die Kosten c_{kl} werden M_∞ gesetzt, wobei M_∞ hinreichend groß ist, damit Objekt l bei der Regret-Berechnung von Mittel k nicht mehr berücksichtigt wird.

In diesem Fall wird im Knoten $t+1$ mit $x_{kl} = 0$ fortgefahren. Dabei ändert sich der Vektor der Kapazitäten nicht ($b^{t+1} := b^t$). Jedoch muss das Regret für das Mittel k neu berechnet werden.

Nun wird wieder das maximale Regret hinsichtlich aller Mittel bestimmt und wie oben beschrieben weiter verfahren. Wird ein Knoten t erreicht, in dem keine Verzweigung mehr zulässig oder möglich ist, jedoch noch nicht jedem Mittel genau ein Objekt zugeordnet wurde, d. h. $\exists i : RG(i) \neq -1$, dann wird entlang des Pfades zum Vorgängerknoten gegangen und eventuell im Knoten t vorgenommene Änderungen in der Kostenmatrix C wieder zurückgesetzt.

Eine zulässige Lösung wurde erreicht, wenn jedes Mittel einem Objekt zugeordnet wurde. Dies ist im Algorithmus daran zu erkennen, dass in einem Knoten alle Regrets gleich -1 sind ($RG(i) = -1\,\forall i$). Dieser Knoten ist ein Blatt des Lösungsbaums. Entlang des Pfades von der Wurzel (Startknoten) zum Blatt kann die Zuordnung eines Mittel zum Objekt und damit die zulässige Lösung X abgelesen werden.

Der Vorteil des Verfahrens ist, dass,wenn die Aufgabe lösbar ist, eine zulässige Lösung gefunden wird. Jedoch ist nicht bekannt, wann diese Lösung gefunden wird. Im Worst Case benötigt der Algorithmus exponentielle Laufzeit, da der vollständige binäre Lösungsbaum $2^{|N|\cdot|M|}$ Knoten besitzt.

Die Lösung, welche mit dem verbesserten Regret-Verfahrens ermittelt wird, ist zulässig, aber im Allgemeinen nicht optimal. Da das Vorgehen ähnlich wie bei der Vogelschen-Approximation ist, kann angenommen werden, dass die Lösung nahe am Optimum liegt und in manchen Fällen bereits optimal ist. Sie dient als Startlösung für ein Verbesserungsverfahren, wie zum Beispiel dem Tabu Search-Verfahren (siehe 4.4.7).

Das eben beschriebene Verfahren wird an einem Beispiel[13] mit 4 Objekten und 10 Mitteln näher erläutert. Es seien die Kostenmatrix C, die Gewichtsmatrix A und der Vektor der Kapazitätsgrenzen b der Objekte wie folgt gegeben:

$$C = \begin{pmatrix} 5 & 22 & 62 & 66 \\ 10 & 15 & 29 & 56 \\ 55 & 25 & 20 & 11 \\ 9 & 12 & 30 & 58 \\ 61 & 24 & 19 & 25 \\ 22 & 4 & 21 & 34 \\ 35 & 19 & 27 & 47 \\ 25 & 22 & 50 & 30 \\ 50 & 17 & 4 & 15 \\ 6 & 16 & 59 & 65 \end{pmatrix} \qquad A = \begin{pmatrix} 0,5 & 0,5 & 0,5 & 0,5 \\ 0,5 & 0,5 & 0,5 & 0,5 \\ 1,0 & 1,0 & 1,0 & 1,0 \\ 1,0 & 1,0 & 1,0 & 1,0 \\ 1,5 & 1,5 & 1,5 & 1,5 \\ 1,5 & 1,5 & 1,5 & 1,5 \\ 1,5 & 1,5 & 1,5 & 1,5 \\ 2,0 & 2,0 & 2,0 & 2,0 \\ 2,5 & 2,5 & 2,5 & 2,5 \\ 5,0 & 5,0 & 5,0 & 5,0 \end{pmatrix} \qquad b = \begin{pmatrix} 5,75 \\ 5,75 \\ 5,75 \\ 5,75 \end{pmatrix}$$

Die Gewichte für ein Mittel i seien hier unabhängig von dem Objekt j. Dies ist daran zu erkennen, dass in jeder Zeile der Matrix A die gleichen Werte stehen. Ebenso sind die Kapazitäten für jedes Objekt gleich. Für die Wurzel des Lösungsbaums gilt $x_{ij} = 0\,\forall i, j$ und $b^0 = b$.

Im ersten Schritt werden die Regrets $RG^0(i)$ für jede Zeile nach (4.32) berechnet. Es ergibt sich der Vektor:

$$RG^0 = \begin{pmatrix} 17 & 5 & 9 & 3 & 5 & 18 & 18 & 3 & 11 & 10 \end{pmatrix}^T$$

[13] Dieses Beispiel entspricht dem Beispiel 1 aus Abschnitt 5.2.2.

Somit existieren zwei Regrets mit maximalen Wert: $RG^0(6) = 18$ und $RG^0(7) = 18$. Die Wahl zwischen Mittel 6 und 7 kann zufällig geschehen. In diesem Beispiel wird das sechste Mittel $(l = 6)$ gewählt. Innerhalb der sechsten Zeile ist c_{62} das kleinste Element $(k = 2)$. Die erste Verzweigung wird somit mit $x_{62} = 1$ bzw. $x_{62} = 0$ vorgenommen. Es wird im Weiteren der Knoten mit $x_{62} = 1$ betrachtet. Dieser Knoten muss nun auf Zulässigkeit geprüft werden. Dabei wird getestet, ob gilt $a_{62} \leq b_2^0$. Diese Bedingung ist erfüllt und die Kapazitätsgrenze wird angepasst: $b_2^1 = b_2^0 - a_{62} = 4,25$. Der zugehörige Regret wird auf -1 gesetzt ($RG^1(6) = -1$), da das Mittel 6 nun zugeordnet wurde. Alle weiteren Regrets ändern sich nicht ($RG^1(i) = RG^0(i)\,\forall i \neq 6$).

Nun wird wieder das maximale Regret ermittelt. Im Beispiel ist dies $RG(7) = 18$. Somit wird das Minimum der Zeile 7 der Matrix C gesucht. Dies entspricht $c_{72} = 19$. Die nächste Verzweigung wird mit der Variable x_{72} vorgenommen. Der Knoten $x_{72} = 1$ wird im Anschluss wieder auf Zulässigkeit geprüft, da das Gewicht a_{72} kleiner als die Kapazitätsgrenze b_2^1 ist, wird $b_2^2 = b_2^1 - a_{72} = 4,25 - 1,5 = 2,75$ gesetzt und mit diesem Knoten fortgefahren. Der neue Regret-Vektor ergibt sich zu $RG^2 = (17|5|9|3|5| - 1| - 1|3|11|10)^T$. Das Verfahren wird analog fortgesetzt.

Der gesamte Lösungsbaum ist in der Abbildung 4.4 dargestellt. Ein erster unzulässiger Knoten entsteht bei der Verzweigung $x_{21} = 1$. In diesem Fall ist das Gewicht $a_{21} = 0,5$ größer als die verbleibende Kapazitätsgrenze $b_1^6 = 0,25$. Dieser Knoten wird gestrichen, die Kosten $c_{21} = M_\infty$ gesetzt und es wird mit der Verzweigung $x_{21} = 0$ fortgefahren. Dabei ändert sich die verbleibenden Kapazitäten nicht, nur der Regret für das Mittel 2 muss neu berechnet werden.

Die Startlösung ist entlang des Pfades mit dem ersten zulässigen Blatt abzulesen. Die Zuordnung lautet:

Mittel		Objekte
{1, 10}	$\rightarrow$	1
{2, 4, 6, 7}	$\rightarrow$	2
{5, 9}	$\rightarrow$	3
{3, 8}	$\rightarrow$	4

Der Zielfunktionswert für diese Zuordnung ergibt sich zu 125. Der Vektor $b^{14} = \{0,25|1,25|1,25|2,75\}$ enthält die noch freien Kapazitäten der Objekte. In dem gewählten Beispiel ist diese Startlösung bereits optimal (vgl. Lösung zum Beispiel 1 im Kapitel 5.2.2).

4.4.6 Savings-Verfahren

Das Savings-Verfahren ist ein Eröffnungsverfahren zur Ermittlung einer Startlösung für des Tourenproblem, welches von Clarke und Wright entwickelt wurde [17]. Als Grundlage wird das aus Abschnitt 4.3.4 beschriebene längenbeschränkte Tourenproblem vorausgesetzt.

Zunächst werden n Pendeltouren $T_i = (i)$ mit $i \in N \backslash \{0\}$ zwischen dem Depot und jedem einzelnen Knoten gebildet (siehe Abbildung 4.5a). Im Anschluss werden je zwei Pendeltouren $T_k = (k)$ und $T_l = (l)$ zu einer neuen Tour $T_{k'}$ vereinigt, indem die zwei Knoten der Touren verbunden werden. Die neue Tour $T_{k'} = (k, l)$ verläuft nun vom Depot zur Stadt k, dann zu l und schließlich wieder zum Depot. In der Abbildung 4.5b wurden die Städte 6 und 7 zu einer Tour verbunden. Im weiteren Verlauf werden jeweils zwei Touren über einen ihrer Endknoten zusammengeschlossen. Endknoten einer Tour $T_i = (i_1, i_2, \ldots, i_r)$ sind ihr erstes und letztes Element, also i_1 und i_r.

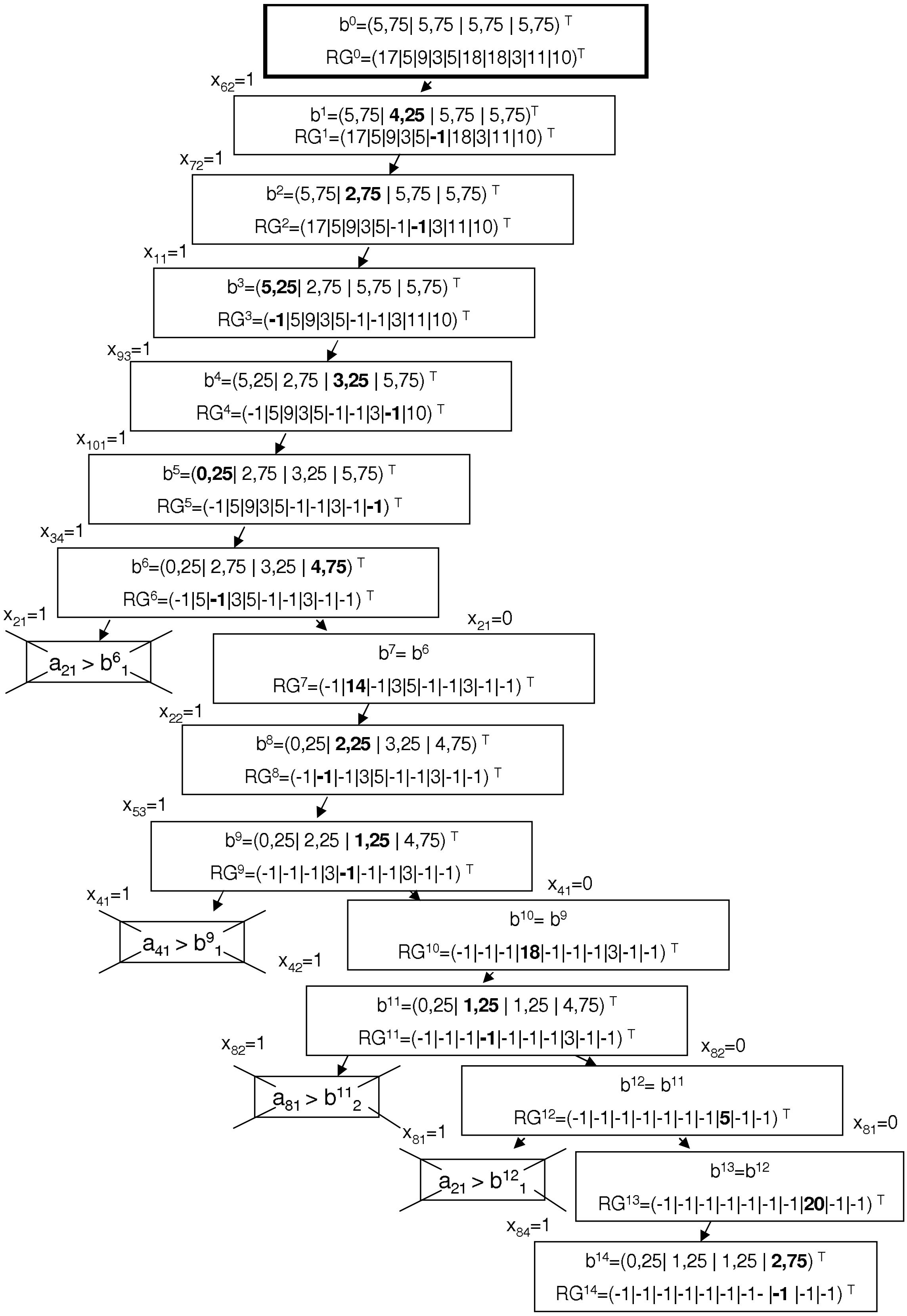


Abbildung 4.4: Lösungsbaum des verbesserten Regret-Verfahrens

Die Wahl zweier Touren, welche zusammengeschlossen werden geschieht über die maximal Ersparnis (engl. Saving), welche durch die Verbindung ihrer Endknoten entsteht. Diese Ersparnis lässt sich für jede Kombination zweier verschiedener Endknoten i, j und 0 als Depot wie folgt berechnen:

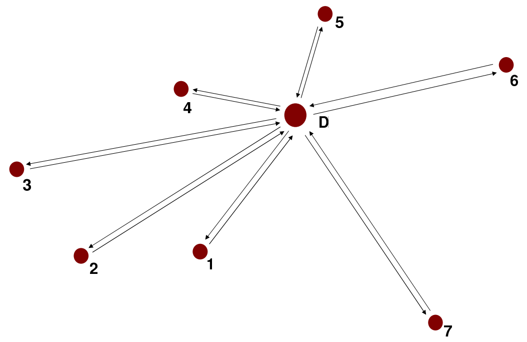
$$s_{ij} = c_{i0} + c_{j0} - c_{ij} \quad \forall i \neq j;\ i, j \in N \backslash \{0\} \tag{4.33}$$

Da beim VRP die Dreiecksungleichung gilt sind alle Savings s_{ij} positiv. Je größer s_{ij} ist, umso größer ist die Ersparnis, welche entsteht, wenn beide Touren zusammengeschlossen werden. Jedoch muss neben der Ersparnis auch die Zulässigkeit des Zusammenschlusses überprüft werden, d. h. ob die Bedingung (4.19) der begrenzten Arbeitszeitkapazität jeder Tour erfüllt ist.

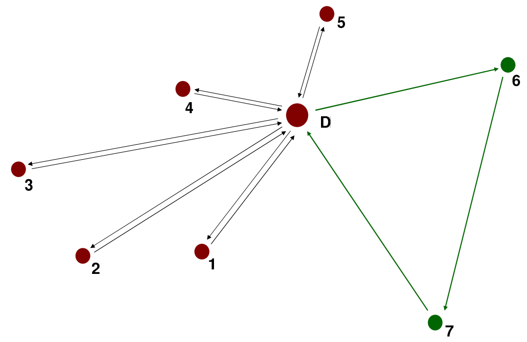
Aus diesem Grund wird der Begriff des zulässigen Savings eingeführt:

Definition 6 (zulässiges Saving)**.** *Ein Saving s_{ij} ist zulässig, wenn beim Zusammenschluss der Touren mit den Endknoten i und j eine neue zulässige Tour entsteht, d. h. diese Tour alle Nebenbedingungen des Ausgangsproblems erfüllt.*

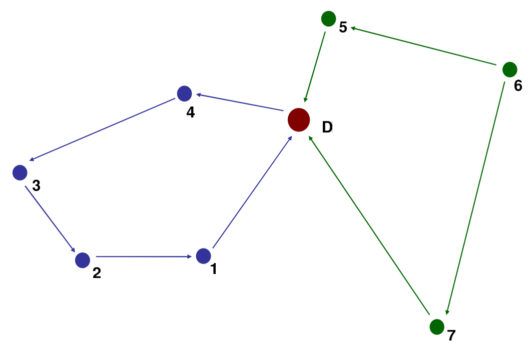
Das Verfahren endet, wenn keine zulässigen Savings mehr existieren, d. h. keine Touren mehr zusammengeschlossen werden können ohne die Arbeitszeitgrenze zu überschreiten. In der Abbildung 4.5c ist für das Beispiel mit sieben Städten das Ergebnis des Saving-Algorithmus dargestellt.



a) Pendeltouren vom Depot D zu sieben verschiedenen Orten



b) Vereinigung der Städte 6 und 7 zu einer Tour



c) Endergebnis des Savings-Algorithmus

Abbildung 4.5: Erläuterung des Savings-Algorithmus

Savings-Algorithmus:

Schritt 1: Bilde alle Pendeltouren T_i. Setze die Menge aller Touren $M = \bigcup_{i \in N \setminus 0} T_i$ und die Menge aller Tourkombinationen $I = \{(i,j) | i,j \in N \setminus \{0\}$ und $i \neq j\}$.

Schritt 2: Berechne die Menge aller Savings $S = \{s_{ij} | (i,j) \in I\}$ mit Hilfe der Gleichung (4.33).

Schritt 3: Sortiere I und S nach der Größe der Elemente in S.

Schritt 4: Sei (k,l) das erste Element aus I. Ermittle die zugehörigen Touren T_i und T_j aus M zu den Endknoten (k,l). Entferne (k,l) aus I.

Schritt 5: Ist das Saving s_{kl} zulässig?

nein Die Touren können nicht zusammengeschlossen werden. Gehe zu Schritt 6.

ja Verbinde T_i und T_j zu einer neuen Tour T_i', in dem die Endknoten i und j verbunden werden. Setze $M = M \setminus \{T_i, T_j\} \cup T_i'$. Gehe zu Schritt 6.

Schritt 6: Ist $I = \emptyset$?

nein Gehe zu Schritt 4.

ja Es sind keine weiteren Zusammenschlüsse möglich. Ausgabe: M

Das Verfahren wird solange ausgeführt bis keine Zusammenschlüsse mehr möglich sind und damit bis keine Touren aufgrund der Arbeitszeitgrenze mehr zusammengeschlossen werden können. Neben der Gesamtfahrzeit wird damit auch die Anzahl der Touren beim Savings-Verfahren minimiert.

4.4.7 Lokale Suchverfahren

Lokale Suchverfahren sind Verbesserungsverfahren. Dabei handelt es sich bei der *Lokalen Suche* um ein Prinzip, worauf viele Verfahren, wie zum Beispiel Tabu Search, Simulated Annealing und 2-opt-Verfahren, beruhen. Bei der Lokalen Suche werden in der *Nachbarschaft* einer zulässigen Startlösung X weitere zulässige Lösungen gesucht, welche einen besseren Zielfunktionswert besitzen.

Innerhalb dieses Abschnitts wird zunächst das Prinzip der Lokalen Suche kurz erläutert. Im Anschluss wird konkret auf eine erweiterte Lösungsstrategie mit Hilfe des Tabu Search Verfahrens und auf eine konkrete Wahl der Nachbarschaft mit dem 2 bzw. 3-opt-Verfahren eingegangen.

Zunächst wird der Begriff der Nachbarschaft einer zulässigen Lösung benötigt:

Definition 7 (Nachbarschaftsfunktion, Nachbarschaft [24])**.** *Es sei $\mathcal{X}$ die Menge der zulässigen Lösungen zu einem Optimierungsproblem. Eine Nachbarschaftsfunktion N ist eine Funktion der Art:*

$$N : \mathcal{X} \to \mathcal{P}(\mathcal{X}) \tag{4.34}$$

mit $\mathcal{P}$ als Potenzmenge. Sie ordnet jeder zulässigen Lösung X eine Teilmenge der Menge aller zulässigen Lösungen zu. Für eine zulässige Lösung $X \in \mathcal{X}$ heißt $N(X)$ die Nachbarschaft von X.

Beim verallgemeinerten Zuordnungsproblem (vZ) aus (4.13) ist die Nachbarschaft einer zulässigen Lösung X zum Beispiel das Vertauschen zweier in X unterschiedlichen Objekten zugeordneten Mittel oder die Verschiebung eines Mittels zu einem anderen Objekt. Wichtig ist dabei, dass eine Lösung, welche durch die Anwendung der Nachbarschaftsfunktion entsteht, wieder zulässig ist, d. h. in diesem Fall die Nebenbedingungen (4.13b) und (4.13c) nicht verletzt werden.

Innerhalb dieser Nachbarschaft wird nun die Lösung $\widetilde{X} \in N(X)$ gesucht, welche einen besseren Zielfunktionswert besitzt. Bei dem verallgemeinerten Zuordnungsproblem ist dies die Lösung mit kleineren Gesamtkosten. Die Lösung $\widetilde{X}$ bildet nun den Ausgangspunkt für weitere Nachbarschaftsuntersuchungen.

Die Lokale Suche endet in einem lokalen Optimum. Ob dieses auch das globale Optimum ist, wird nicht erkannt. Da das Verfahren immer nur *„lokal"*, also in der Nachbarschaft, nach einer besseren Lösung sucht, ist die Güte der Lösung stark von der Startlösung abhängig. Eine Erweiterung dieser Methode, bei der nicht nur ein Nachbar mit besserem Zielfunktionswert einen neuen Ausgangspunkt bildet, ist das *Tabu Search* Verfahren.

Lösungsstrategie: Tabu Search Verfahren

Das Problem lokaler Suchverfahren ist, dass diese in einem lokalen Minimum enden und sich von diesem nicht entfernen. Das Tabu Search Verfahren ist ein Iterationsverfahren und endet nicht bei dem Erreichen eines lokalen Minimum, sondern sucht weiter nach besseren Lösungen. Dazu wird der Suchraum im Unterschied zur Lokalen Suche vergrößert, also die Menge an zulässigen Lösungen, welche untersucht werden.

Dabei ist der Unterschied zur einfachen Lokalen Suche, dass nicht nur die Lösung mit einem besseren Zielfunktionswert als neuen Ausgangspunkt dienen kann, sondern auch schlechte in Frage kommen. Die Lösung mit den besten Zielfunktionswert kann *tabu* sein. Ein Nachbar ist *tabu* und kann nicht als neuer Ausgangspunkt dienen, wenn dieser in der *Tabuliste* enthalten ist. Die Tabuliste enthält die letzten k Lösungen der Iteration. Mit Hilfe der Tabuliste soll verhindert werden, dass immer zum lokalen Optimum zurückgekehrt wird, d. h. das Verfahren sich zyklisch wiederholt[14]. Als neuer Ausgangspunkt dient der Nachbar, welcher den besten Zielfunktionswert besitzt und nicht in der Tabuliste enthalten ist.

Das Verfahren endet, wenn alle Nachbarn in der Tabuliste enthalten sind oder nach einer vorher festgelegten Anzahl an Iterationsschritten. Die Lösung, welche im Verlauf den besten Zielfunktionswert besitzt, wird ausgegeben.

Die Länge k der Tabuliste muss vor Beginn des Verfahren festgelegt werden. Je größer k ist, desto größer ist der Rechen- und Speicheraufwand. Jedoch werden Zyklen der Länge kleiner als k verhindert. Die Wahl von k entscheidet wie weit von einer Startlösung aus der Algorithmus sich „entfernt".

Beim verallgemeinerten Zuordnungsproblem sei die Startlösung X gegeben. Diese kann zum Beispiel mit dem verbesserten Regret-Verfahren (siehe Abschnitt 4.4.5) erzeugt werden. Die Nachbarschaft dieser Lösung wurde bereits im vorangegangen Abschnitt definiert. Es sei $N(X)$ die Menge aller Nachbarn, welche durch Vertauschung zweier Mitteln zweier unterschiedlicher Objekte oder durch Zuweisung eines Mittels zu einem anderen Objekt entsteht. Zudem wird der Vektor $Z_{N(X)}$ mit den zugehörigen Zielfunktionswerten berechnet.

Ausgehend von der Startlösung X_0, welche zunächst die beste Lösung X_b ist, wird nun zunächst der Nachbar gesucht, welcher den kleinsten Zielfunktionswert besitzt. Besitzt dies Lösung X_1 einen kleiner Zielfunktionswert als die Startlösung, wird $X_b = X_1$ gesetzt. Die Startlösung wird nun der Tabuliste $T = \{X_0\}$ hinzugefügt. Nun wird wieder die Nachbarschaft von X_1 nach der Lösung mit dem kleinsten Zielfunktionswert untersucht. Ist dies die Lösung X_0, dann ist diese tabu

[14] Genauer gesagt, werden Zyklen der Länge kleiner als k verhindert

und stattdessen wird die Lösung mit dem zweit kleinsten Zielfunktionswert als X_2 verwendet. Der Zielfunktionswert von X_2 wird nun wieder mit X_b vergleichen. Ist er kleiner, wird X_2 neue beste Lösung. Die Lösung X_1 wird der Tabuliste T angefügt. Das Verfahren wird nun fortgeführt. Hat die Tabuliste die Länge k erreicht, so wird immer das zuerst hinzugefügte Element entfernt.

Erweiterungen des Tabu Search-Verfahrens können unter anderem in [23] nachgelesen werden. Der grundlegende Algorithmus von Tabu Search für Minimierungsprobleme wird im Folgenden dargelegt:

Tabu-Search-Algorithmus

Initialisierung: Es sei X_0 eine zulässige Startlösung. Setze die Tabuliste $T = \emptyset$, $i = 0$ und die bisher beste Lösung $X_b = X_0$ mit dem Zielfunktionswert $z(X_b)$. Zudem sei die maximale Anzahl an Iterationsschritten l.

Schritt 1: Suche die Lösung $X_{i+1} \in N(X_i)$ mit dem kleinsten Zielfunktionswert, für die gilt $T \cap X_{i+1} = \emptyset$

Schritt 2: Kann kein X_{i+1} gefunden werden?

- nein Ist $|T| = k$, dann entferne das erste Element. Setze $T = T \cup X_{i+1}$ und $i = i + 1$. Gehe zu Schritt 3.
- ja Die gesamte Nachbarschaft wurde bereits untersucht. Ausgabe: X_b

Schritt 3: Gilt $z(X_b) > z(X_i)$?

- nein Gehe zu Schritt 4.
- ja Setze $X_b = X_i$.

Schritt 4: Ist $i \geq l$?

- nein Gehe zu Schritt 1.
- ja Es wurden l Iterationen durchgeführt. Ausgabe: X_b

4.4.8 2-opt-Verfahren

Das 2-opt-Verfahren ist ebenso ein lokales Suchverfahren, welches jedoch nicht den Ablauf des Algorithmus näher beschreibt, sondern die Nachbarschaftsfunktion. Die Definition der Nachbarschaftsfunktion ist sehr allgemein gefasst und speziell für die einzelnen Probleme festzulegen. So sind zum Beispiel bei dem verallgemeinerten Zuordnungsproblem die Nachbarn zulässige Lösungen, welche durch Vertauschung von zwei Mitteln zwischen zwei verschieden Objekten oder der Zuordnung eines Mittel zu einem anderen Objekt entstehen.

Das 2-opt-Verfahren tritt meistens in Verbindung mit graphentheoretischen Problemen auf. Die Nachbarschaftsfunktion definiert sich über die Vertauschung zweier verschiedene Kanten einer Lösung, so dass die neu entstandene Lösung wieder zulässig ist. Die *2-opt-Nachbarschaft* ist die Menge aller möglichen Vertauschungen.

Beim Rundreiseproblem bedeutet dies: Seien $e_1 = \{u, v\}$, $e_2 = \{w, x\}$ zwei Kanten eines Hämiltonkreises, welche keine gemeinsamen Endknoten besitzen, dann ergibt sich bei Anwendung

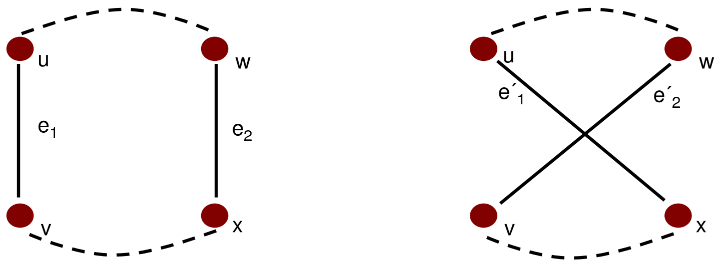


Abbildung 4.6: Veranschaulichung einer 2-opt-Nachbarschaft. Die gestrichelten Linien deuten an, dass die Knoten $u \rightarrow w$ bzw. $v \rightarrow x$ über einen Weg miteinander verbunden sind.

der Nachbarschaftsfunktion ein neuer Hämiltonkreis mit den neuen Kanten $e'_1 = \{u, x\}$ und $e'_2 = \{w, v\}$ durch Austausch der Endknoten. Verdeutlicht wird dies in der Abbildung 4.6.

Die Mächtigkeit der 2-opt-Nachbarschaft beim TSP beträgt $\frac{n \cdot (n-3)}{2}$, wenn die Anzahl der Knoten $n \geq 3$ beträgt. Jedoch wird beim 2-opt-Verfahren im Allgemeinen nicht die gesamte Nachtbarschaftsmenge erzeugt, sondern eine Nachbar gewählt, welcher eine kürzere Rundreise als die Ausgangsrundreise besitzt.

Das 2-opt-Verfahren kann auch mit weiteren Suchverfahren kombiniert werden, welche eine andere Auswahlstrategien der Nachbarn besitzen, wie z. B. das Tabu Search Verfahren. Ein einfacher 2-opt-Algorithmus ist im Folgenden für das Rundreiseproblem (TSP) zusammengefasst:

2-opt-Verfahren

Initialisierung: Gegeben sei die Entfernungsmatrix $C = (c_{ij})_{n,n}$ und eine Startlösung $X_0 = (x^0_{ij})_{n,n}$. Setze die Kantenmengen $K_1 = \emptyset$, $K_2 = \emptyset$, die bisher beste Lösung $X_b = X_0$ und den Iterationszähler $p = 0$.

Schritt 1: Wähle eine Kante $e = (i, j) \notin K_1$ mit $x^p_{ij} = 1$. Setze $K_1 = K_1 \cup \{(i, j)\}$ und $K_2 = K_1$.

Schritt 2: Wähle eine weitere Kante $f = (k, l) \notin K_2$ mit $x^p_{kl} = 1$ und $k \neq i$, $k \neq j$.
$K_2 = K_2 \cup \{(k, l)\}$

Schritt 3: Gilt $c_{ij} + c_{kl} > c_{il} + c_{kj}$?

nein Gilt $|K_2| < n - 1$?

nein Alle anderen Kanten zu (i, j) wurden bereits untersucht. Setze $K_2 = \emptyset$ und gehe zu Schritt 4.

ja Gehe zu Schritt 2.

ja Setze $p = p + 1$. Setze $X_p = X_{p-1}$ mit $x^p_{ij} = 0$, $x^p_{kl} = 0$ und $x^p_{il} = 1$, $x^p_{kj} = 1$.

Setze $X_b = X_p$ und $K_1 = \emptyset, K_2 = \emptyset$. Gehe zu Schritt 5.

Schritt 4 Gilt $|K_1| < n - 2$?

nein Es konnte keine besser Lösung mit dem 2-opt-Verfahren gefunden werden.
Ausgabe: X_b

ja Gehe zu Schritt 1.

Schritt 5: Ist $p \geq$ Anzahl der maximalen Iterationsschritte?

nein Gehe zu Schritt 1.

ja Es wurden die maximale Anzahl an Iterationsschritten durchgeführt. Ausgabe: X_b

Das 2-opt-Verfahren kann ebenso auf das Tourenprobleme angewandt werden. Erfolgt die Wahl der zwei Kanten innerhalb einer Tour, dann wird dies als toureninterne Verbesserung bezeichnet. Der Ablauf ist analog zum Rundreiseproblem. Es findet eine Veränderung in der Reihenfolge der Städte innerhalb einer Tour statt.

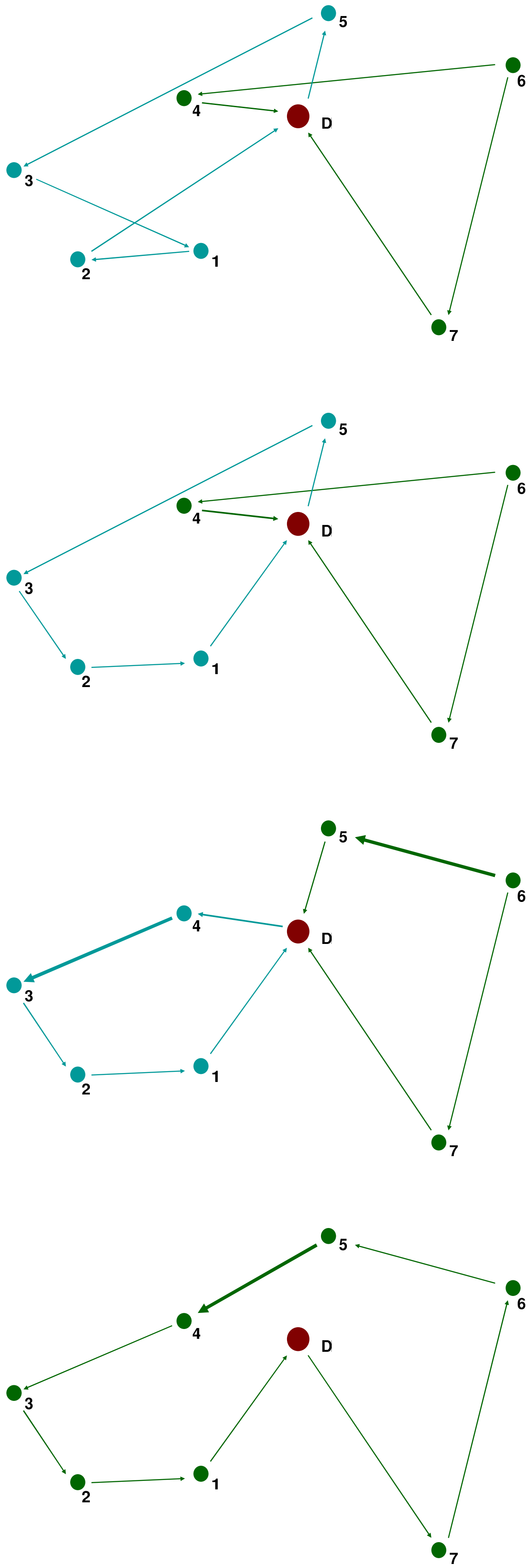
Neben der toureninternen kann auch eine tourenübergreifende Verbesserung vorgenommen werden. Dabei werden die zwei Kanten aus zwei verschiedenen Touren gewählt. bei tourenübergreifenden Verbesserungen muss zusätzlich die Zulässigkeit der Vertauschung geprüft werden, also ob die sich neu ergebenen Touren alle Nebenbedingungen erfüllen. Die Städte der beiden Touren werden neu aufgeteilt und damit auch eine neue Reihenfolge festgelegt. Die beiden neu entstandenen Touren müssen die Arbeitszeitgrenze einhalten.

In der Abbildung 4.7 werden von einer Startlösung aus drei verschiedene Verbesserungsschritte angegeben. Der erste Schritt wählt zwei Kanten aus einer Tour und es ergibt sich eine toureninterne Verbesserung. Im zweiten Schritt werden zwei Kanten aus zwei unterschiedlichen Touren gewählt. Die Städte 4 und 5 werden jeweils der anderen Tour zugeordnet. Eine Besonderheit tritt auf, wenn beide Kanten das Depot als einen Endknoten besitzen, dann verschmelzen bei der Anwendung beide Touren miteinander. Ein Beispiel für die Verschmelzung ist in der Abbildung 4.7c) aufgezeigt.

Verallgemeinerung: r-opt-Verfahren

Eine Verallgemeinerung des 2-opt-Verfahren ist das r-opt-Verfahren, dabei findet eine Austausch von r Kanten statt. Je größer r gewählt wird, desto so mehr Möglichkeiten gibt es, diese zu kombinieren. Dabei ist immer darauf zu achten, dass eine Kombination wieder eine zulässige Lösung ist. Im Beispiel der Tourenplanung besteht die Bedingung, dass die neue Kombination der Kanten wieder zulässige Touren ergibt. Auch hier wird zwischen toureninterner und tourenübergreifender Verbesserung analog wie beim 2-opt-Verfahren unterschieden. Das oben beschriebene 2-opt-Verfahren kann auf das r-opt-Verfahren übertragen werden.

Mit der Anzahl der Kombinationsmöglichkeiten steigt auch der Aufwand des Verfahrens, da für jede mögliche Kombination getestet werden muss, ob diese eine zulässige Tour ist und wenn dies gegeben ist, muss der Zielfunktionswert berechnet werden. Neben $r = 2$ wird $r = 3$ in der Literatur oft verwendet, welche aus diesem Grund auch innerhalb dieser Arbeit im weiteren Verlauf zur Anwendung kommen. Neben der einfachen Vorgehensweise gibt es viele Erweiterungen, wie zum Beispiel das OR-Verfahren oder das Lin-Kernighan-Verfahren. Als weiterführende Literatur wird hier [15] empfohlen.



Startlösung

a) Wahl der Kanten (3,1) und (2,0) zur toureninterne Verbesserung

b) Wahl der Kanten (6,4) und (5,3) zur tourenexterne Verbesserung

c) Wahl der Kanten (0,4) und (5,0) zur Tourenverschmelzung

Abbildung 4.7: Verbesserungsschritte des 2-opt-Verfahrens

5 Alternative Herangehensweise

Die im Kapitel 3 beschriebene Herangehensweise mit der Trennung in Auftragseinplanung und Vordisposition ist analog zum Vorgehen des Dispatchers und erfolgt somit wie in der Praxis. Die Implementierung eines Softwaretools nach dem im Kapitel 3.2 beschriebenen Optimierungszielen wird von der Firma *Fuzzy-Logic-Systems* umgesetzt und soll aus diesem Grund nicht näher betrachtet werden.

In diesem Abschnitt wird eine alternative Herangehensweise erläutert und dieser Ansatz modelliert. Es wird nicht mehr zwischen Auftragseinplanung und Vordisposition in der genannten Form unterschieden. Im ersten Ansatz werden die Aufträge zunächst den Tagen und im Anschluss den Einheiten zugewiesen. Eine alternative Herangehensweise ist die Umkehrung: Zunächst werden alle Aufträge den Einheiten zugewiesen und im Anschluss findet eine Zuweisung zu den Tagen statt. In den folgenden Abschnitten werden diese Herangehensweise näher erläutert.

5.1 Lösungsstrategien

Die betrieblichen Ziele aus 3.1.1 gelten auch hier, d. h. der neue Ansatz wird hinsichtlich dieser Ziele modelliert. Die Begriffe Auftragseinplanung und Vordisposition können nun nicht mehr verwendet werden, stattdessen werden die Begriffe *Auftragszuordnung* und *Tagestouren* eingeführt.

Im Folgenden wird eine Einheit mit Monteur bezeichnet, damit keine Verwechslung zum mathematischen Begriff der Einheit entsteht. Zudem wird der Begriff *Auftrag* für den Hauptvorgang mit eventuell zugehörigen Nebenvorgängen, welche die Monteure bearbeiten, verwendet. Der Hauptvorgang bildet im Wesentlichen den Auftrag ab (siehe 5.3.2).

Auftragszuordnung Jedem Auftrag wird ein Monteur zugeordnet.

Die Auftragszuordnung erfolgt unter Beachtung der Ressourcen und der Auslastung der Monteure, d. h. alle Monteure sollten ein nahezu gleiches Arbeitszeitvolumen an Aufträgen erhalten. Das Volumen eines Auftrages berechnet sich aus der Summe der Plandauern der einzelnen Vorgänge. Die Ausführungszeiträume spielen zunächst keine Rolle.

Zunächst werden die Sonderfälle in den Auftragstypen nicht berücksichtigt, diese werden im Anschluss im Abschnitt 5.4.1 diskutiert. Sonderfälle sind zum Beispiel Auftragstypen mit dem Bedarf von mehr als einen Mitarbeiter.

Tagestouren Die Aufträge werden in ihre Vorgänge unterteilt und jedem Vorgang wird der Planbeginn zugeordnet. Somit erhält jeder Monteur seine Tagestouren.

Im Anschluss an die Auftragszuordnung können bei den Tagestouren alle zeitlichen Bedingungen und Ziele berücksichtigt werden. Zum Beispiel sollte der Ausführungszeitraum eines Auftrages und seine Priorität nun Beachtung finden.

Bei den Tagestouren werden im Anschluss Fahrpläne für jeden einzelnen Monteur ermittelt. Die zur Verfügung stehenden Aufträge sind die, welche bei der Auftragszuordnung dem Monteur zugewiesen wurden. Jedem Tag werden nun Aufträge zugeordnet und eine Reihenfolge der Abarbeitung festgelegt. Dabei sollte die Fahrstrecke möglichst minimal sein.

Auch hier wird zunächst ein vereinfachtes Ausgangsproblem angenommen und Lösungsideen für die Spezialfälle, wie beispielsweise Mehrtageseinsätzen, erst im Abschnitt 5.4.1 angesprochen.

5.2 Auftragszuordnung

Es sei $M = \{m_1, m_2, \ldots, m_{|M|}\}$ die Menge der Aufträge und $N = \{n_1, n_2, \ldots, n_{|N|}\}$ die Menge der Monteure für einen Planungszeitraum T. Zudem sei p_{m_i} die Summe der Plandauern aller Vorgänge des Auftrages m_i. Die Luftlinienentfernung zwischen dem Ausführungsort vom Auftrag m_i und der Startadresse vom Monteur n_j wird mit $d(m_i, n_j)$ bezeichnet.

Im Kapitel 3.1.1 wurden alle betrieblichen Ziele erläutert. Innerhalb der Auftragszuordnung sollen nun einige dieser Ziele verfolgt werden. Ein wichtiges Kriterium ist die hohe und vor allem gleichmäßige Auslastung der Monteure. Dies wird erzielt, indem allen Monteure ein möglichst gleiches Arbeitszeitvolumen zugeordnet bekommen.

Des Weiteren sollen die Ressourcen als Nebenbedingungen im Modell mit berücksichtigt werden. Eine Erweiterung des Modells unter dem Aspekt des minimalen Ressourcenverbrauchs wird später im Abschnitt 5.4.1 gegeben. Zudem wird zunächst angenommen, dass ein Auftrag vollständig von einem Monteur bearbeitet wird. Die Erweiterung im Modell, dass Vorgänge in einem Auftrag von verschiedenen Monteuren bearbeitet werden, wird ebenso im Abschnitt 5.4.1 angesprochen.

Indirekt kann das Ziel der Minimierung der Fahrzeit berücksichtigt werden. Dies geschieht durch die Zuordnung der Aufträge zu den Monteuren in Abhängigkeit ihrer Entfernungen, d. h. ein Auftrag soll möglichst dem zulässigen Monteur zugeordnet werden, welcher die geringste Luftlinienentfernung zum Ausführungsort des Auftrages besitzt. Ob eine Zuordnung zu dem Monteur zulässig ist, entscheiden die Nebenbedingungen, wie zum Beispiel die Ressourcen.

Alle weiteren betrieblichen Ziele beziehen sich auf die Ausführungszeit und müssen deshalb bei der Auftragszuordnung nicht berücksichtigt werden. Zudem wird zunächst die Bedingung gestellt, dass alle Monteure über den gesamten Zeitraum zur Verfügung stehen und somit eine gleiche Arbeitszeitkapazität besitzen. Auf die Erweiterung des Modells bei Nichterfüllung dieser Bedingung wird ebenfalls in 5.4.1 eingegangen.

Zusammengefasst ist bei der Auftragszuordnung nun eine Zuordnung der Monteure gesucht, so dass gilt:

AZ(a) Die Gesamtluftlinienentfernung zwischen den Ausführungsorten aller Aufträge und den Startadressen der Monteure ist minimal.

AZ(b) Alle Monteure besitzen eine nahezu gleiche Gesamtplandauer der zugeordneten Aufträge.

AZ(c) Jeder Auftrag wird genau einem Monteur zugeordnet.

AZ(d) Einem Monteur n_j wird nur dann der Auftrag m_i zugeteilt, wenn dieser die erforderlichen Ressourcen (Berechtigungen, Ausrüstung und Fähigkeiten) besitzt.

5.2.1 Modellierung

Das Ziel dieses Kapitels ist, das Problem der Auftragszuordnung so zu modellieren, dass es mit Hilfe bekannter Optimierungsmethoden effizient gelöst werden kann. Der Begriff der Auftrags-***zuordnung*** beinhaltet bereits eine erste Idee: die Betrachtung des Problems als Zuordnungsproblem.

Im Abschnitt 4.3.2 wurde das mathematische Modell des linearen Zuordnungsproblems angegeben. In dem gegebenen Problem entsprechen die Aufträge den Mitteln und die Monteure den Objekten. Die Zielfunktion ist die Minimierung der Summe der Entfernungen $c_{ij} = d(m_i, n_j)$ zwischen den Aufträgen und den Startadressen der Monteure mit der Zuordnungsvariablen x_{ij}, wobei gilt:

$$x_{ij} = \begin{cases} 1 & \text{, wenn der Auftrag } m_i \text{ dem Monteur } n_j \text{ zugeordnet wird} \\ 0 & \text{, sonst.} \end{cases} \tag{5.1}$$

Die Forderung, dass jedem Monteur genau ein Auftrag zugeordnet wird, ist bei dem gegebenen Problem jedoch nicht erfüllbar, da die Anzahl der Aufträge weit aus höher ist als die Anzahl der Monteure. Es gilt $|M| <<< |N|$. Aus diesem Grund wird eine Verallgemeinerung des Zuordnungsproblems gesucht.

Das verallgemeinerte Zuordnungsproblem (4.13) erfüllt genau diese Bedingung, d. h. einem Monteur können mehrere Aufträge zugeordnet werden und jeder Auftrag wird genau einem Monteur zugeordnet. Die im Modell (4.13) benötigte Größe a_{ij} lässt sich durch die Plandauer p_{m_i} ersetzen. Zudem erhält jeder Monteur eine maximale Gesamtarbeitszeit b_j, d. h. die Summe der Plandauern der Aufträge, welche dem Monteur n_j zugewiesen werden, darf b_j nicht überschreiten. Eine Bedingung an die Auftragszuordnung ist, dass jeder Monteur eine nahezu gleiche Summe an Plandauern erhält. Da alle Monteure zunächst eine gleiche Arbeitszeitkapazität besitzen, kann eine einheitliche Grenze $b = b_j \ \forall j$ gewählt werden. Wenn der Parameters b über die Mittlung der Plandauer aller Aufträge

$$b = \frac{1}{|N|} \sum_{i=1}^{|M|} p_{m_i} \tag{5.2}$$

berechnet wird, dann ergibt sich das folgende Modell:

(AZ)

$$\text{ZF:} \qquad \sum_{i=1}^{|M|}\sum_{j=1}^{|N|} c_{ij}x_{ij} \rightarrow min \tag{5.3a}$$

$$\text{unter den NB:} \qquad \sum_{j=1}^{|N|} x_{ij} = 1 \qquad i = 1,\ldots,|M| \tag{5.3b}$$

$$\sum_{i=1}^{|M|} p_{m_i}x_{ij} = b \qquad j = 1,\ldots,|N| \tag{5.3c}$$

$$x_{ij} \in \{0,1\} \qquad i = 1,\ldots,|M|,\ j = 1,\ldots,|N| \tag{5.3d}$$

Die Zielfunktion spiegelt die Forderung *AZ(a)* wider. Die erste Nebenbedingung gewährleistet, dass jeder Auftrag genau einem Monteur zugeordnet wird (*AZ(c)*). Die zweite Nebenbedingung fordert die gleichmäßige Aufteilung der Plandauern auf die Einheiten (*AZ(b)*) und die binäre Variable x_{ij} bewirkt die Eindeutigkeit der Zuordnung.

Eine offenen Bedingung ist die Zulässigkeit einer Zuordnung (Forderung *AZ(d)*). Einem Monteur soll nur dann ein Auftrag zugeordnet werden, wenn dieser über die gültigen Ressourcen verfügt. Dies kann mit Hilfe der Zielfunktionskoeffizienten c_{ij} berücksichtigt werden:

$$c_{ij} = \begin{cases} d(m_i, n_j) & \text{, wenn der Monteur } n_j \text{ die Ressourcen des Auftrages } m_i \text{ erfüllt} \\ M_\infty & \text{, sonst} \end{cases} \tag{5.4}$$

Dabei wird M_∞ hinreichend groß gewählt, so dass eine solche Zuordnung stark entgegen der Optimierungsrichtung wirkt und damit nicht optimal ist. Folglich wird bei Nichtvorhandensein der Ressourcen für Auftrag m_i beim Monteur n_j die Zuordnungsvariable $x_{ij} = 0$ sein.

Jedoch ist im Allgemeinen bei praktischen Problem der Restriktionsbereiches des Modells (5.3) leer. Ursache dafür ist die zweite Nebenbedingung und die binäre Forderung an die Variablen. Eine Gleichheit der Summe der Plandauern für jeden Monteur mit der gemäß (5.2) berechnete mittleren Plandauer b kann in den meisten Fällen nicht erfüllt werden. Offensichtlich ist dies, wenn beispielsweise die mittlere Plandauer b nicht ganzzahlig ist, jedoch alle Plandauern $p_{m_i} \in \mathbb{N}$ sind. Die Summe natürlicher Zahlen ist immer natürlich und damit kann für keinen Monteur die zweite Nebenbedingung in (5.3) erfüllt werden.

Es ergeben sich zwei Möglichkeiten um dieses Problem zu beheben. Zum einen kann der Parameter b um einen weiteren Parameter $\delta > 0$ vergrößert und damit die Gleichungsnebenbedingung (5.3c) in eine Ungleichung umgeformt werden. Zum anderen könnte eine Relaxation der Binärforderung durch die Nichtnegativitätsbedingung vorgenommen werden.

Bei Anwendung der ersten Möglichkeit durch Anpassung der rechten Seite ergibt sich aus der zweiten Nebenbedingung:

$$\sum_{i=1}^{|M|} p_{m_i}x_{ij} \leq b + \delta \quad j = 1\ldots|N| \tag{5.5}$$

Die zweite Alternative, die Lösung mittels Relaxation, bedeutet zunächst die Binärforderung zu vernachlässigen, das relaxierte Problem als LOA zu lösen und im Anschluss die optimale Lösung so zu modifizieren, dass eine eindeutige Zuordnung der Aufträge zu den Monteuren hergestellt wird. Diese beiden Möglichkeiten werden im Folgenden erläutert.

5.2.2 Lösung mittels Anpassung der rechten Seite

Das Modell (AZ) mit der modifizieren zweiten Nebenbedingung (5.5) ist ein verallgemeinertes Zuordnungsproblem. Die Theorie wurde im Kapitel 4.3.2 mit den möglichen Lösungsmethoden angegeben.

Die Mittel entsprechen den Aufträgen und die Objekte den Monteuren. Die Gewichte a_{ij} entsprechen den Plandauern p_{m_i} und die Kapazitätsgrenzen b_j der Arbeitszeitkapazität $b + \delta\, \forall j$. Die Lösungen, welche mit dem verbesserten Regret-Verfahren und Tabu Search Verfahren ermittelt werden, sind Näherungslösungen und liefern somit eine fast optimale Lösung.

Mit Hilfe des Parameters δ in (5.5) soll sicher gestellt werden, dass der Restriktionsbereich nicht mehr leer ist. Die Wahl des Wertes für δ ist jedoch nicht trivial. Wird der Wert für δ zu klein gewählt, könnte $b + \delta$ immer noch zu klein sein und der Restriktionsbereich ist möglicherweise wiederum leer. Wird im Gegenzug δ zu groß gewählt, so kann ein Ungleichgewicht in der Gesamtplandauer bei der Verteilung der Aufträge entstehen. So könnte in einer zulässigen Lösung allen bis auf einem Monteur eine Gesamtplandauer von $b + \delta$ zugeordnet werden und für den letzten Monteur würde nur noch eine Gesamtplandauer von $b - (|N| - 1)\delta$ zur Verfügung stehen. Dieses Lösung kann im Bezug auf die Entfernungen fast optimal sein, jedoch verstößt sie gegen die Forderung *AZ(b)*. Das beschriebene Problem wird am folgenden Beispiel deutlich.

Beispiel 1: Gegeben seien 4 Monteure $N = \{n_1, n_2, n_3, n_4\}$ und die Auftragsmenge $M = \{m_1, m_2, \ldots, m_{10}\}$ mit der Plandauer p_{m_i} aus der Tabelle 5.1. Durch die Mittlung (5.2) der Plandauern ergibt sich für $b = 4,25\, h$. Die Luftlinienentfernungen zwischen den Ausführungsorten der Aufträgen und den Startadressen der Monteure sind ebenso aus der Tabelle 5.1 abzulesen. Zunächst sollen alle Monteure die benötigten Ressourcen für alle Aufträge besitzen. Damit ergeben sich die Zielfunktionskoeffizienten c_{ij} direkt aus der Entfernungstabelle:

	Entfernungen in km				Plandauer in h
Monteure / Aufträge	n_1	n_2	n_3	n_4	p_{m_i}
m_1	5	22	62	66	$0,5$
m_2	10	15	29	56	$0,5$
m_3	55	25	20	11	$1,0$
m_4	9	12	30	58	$1,0$
m_5	61	24	19	25	$1,5$
m_6	22	4	21	34	$1,5$
m_7	35	19	27	47	$1,5$
m_8	25	22	50	30	$2,0$
m_9	50	17	4	15	$2,5$
m_{10}	6	16	59	65	$5,0$

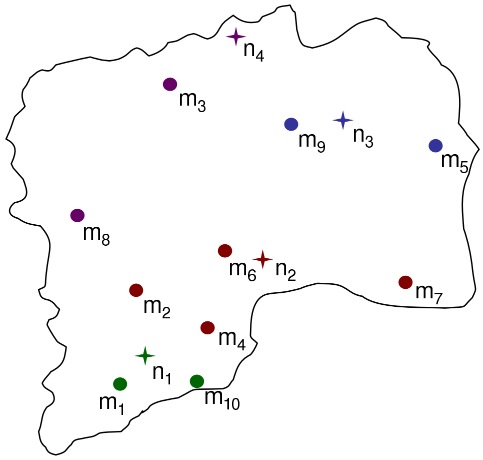
Tabelle 5.1: Entfernungs- und Plandauertabelle für das Beispiel 1

a)

$\delta = 0,5h$, $b + \delta = 4,75h$
Restriktionsbereich ist leer

b)

$\delta = 1,5h$, $b + \delta = 5,75h$		
Monteur	zugeordnete Aufträge	Summe der Plandauern
n_1	m_1, m_{10}	$5,5\,h$
n_2	m_2, m_4, m_6, m_7	$4,5\,h$
n_3	$m_5, m_9,$	$4\,h$
n_4	$m_3, m_8,$	$3\,h$
Zielfunktionswert: 125 km		



c)

$\delta = 0,75h$, $b + \delta = 5h$		
Monteur	zugeordnete Aufträge	Summe der Plandauern
n_1	m_{10}	$5\,h$
n_2	m_1, m_2, m_4, m_6, m_7	$5\,h$
n_3	$m_5, m_9,$	$4\,h$
n_4	m_3, m_8	$3\,h$
Zielfunktionswert: 142 km		

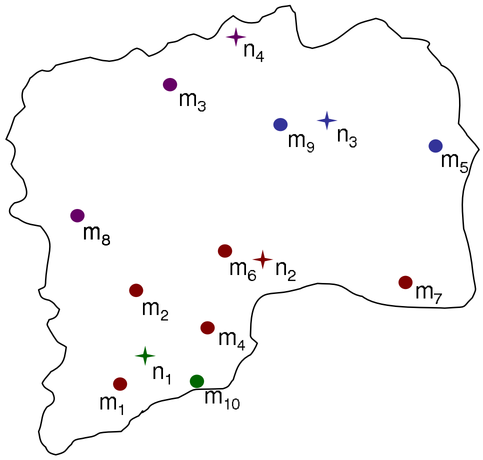


Abbildung 5.1: Beispiel 1 mit Aufträgen (Kreise) bzw. Monteuren (Kreuze) und mit der optimalen Zuordnung in der Tabelle

Wird nun $\delta = 0,5\,h$ angesetzt, ist ersichtlich, dass der Auftrag m_{10} nicht zugeordnet werden kann, da die Grenze $4,75\,h$ kleiner als die Plandauer $p_{m_{10}}$ ist. Wird $\delta = 1,5\,h$ bzw. $\delta = 0,75\,h$ gewählt, erweisen sich die Zuordnungen, welche in der Abbildung 5.1 dargestellt sind, hinsichtlich der Luftlinienentfernung als optimal[1].

Im der Tabelle 5.1b) wird deutlich, dass die Zuordnung bei der Wahl von $\delta = 2\,h$ hinsichtlich der Luftlinienentfernung zwar optimal ist, jedoch keine gleichmäßige Verteilung der Plandauern vorliegt. Die Monteure n_1, n_2 und n_3 sind voll ausgelastet und dem Monteur n_4 werden nur wenige Aufträge mit geringer Plandauer zugeordnet. Eine bessere Verteilung der Arbeitszeiten ergibt sich, wenn $\delta = 0,75\,h$ gewählt wird (vergleiche Tabelle 5.1c)), jedoch verschlechtert sich die Gesamtentfernung um $17\,km$.

Bei diesem selbst gewählten Beispiel ist zu beachten, dass die Anzahl der Aufträge und Monteure sehr gering ist und damit sich der Effekt bei der schlechten Wahl von δ verstärkt. Ursache dafür ist der Auftrag m_{10} mit der relativ großen Plandauer von 5 Stunden.

[1]Ermittlung der Lösung mit Hilfe des Excel Solvers.

5.2.3 Lösung mittels Relaxation

Das Zuordnungsproblem ist ein spezielles Transportproblem bei dem unter anderen die Variablen x_{ij} binär sind. Im Gegenzug ist das Transportproblem ein lineares Problem. Wird nun das verallgemeinerte Zuordnungsproblem ohne Binärforderung betrachtet, ergibt sich ein Transportproblem. Dies wird im Folgenden hergeleitet. Die Idee dieses Vorgehens entstammt der Dissertation [26] und wird auf das Problem der Auftragszuordnung angewendet.

Zunächst wird die binäre Forderung an die Variable x_{ij} durch die NNB $x_{ij} \geq 0$ relaxiert. Es ergibt sich eine LOA als relaxiertes Problem. Da nun x_{ij} nicht mehr ganzzahlig sein muss, kann die zweite Nebenbedingung im Modell (5.3) als eine Gleichungsbedingung bestehen bleiben. Es ergibt sich das folgende Modell:

(AZ_{R1})

ZF: $$\sum_{i=1}^{|M|}\sum_{j=1}^{|N|} c_{ij}x_{ij} \rightarrow min \tag{5.6a}$$

unter den NB: $$\sum_{j=1}^{|N|} x_{ij} = 1 \qquad i = 1, \ldots, |M| \tag{5.6b}$$

$$\sum_{i=1}^{|M|} p_{m_i}x_{ij} = b \qquad j = 1, \ldots, |N| \tag{5.6c}$$

$$x_{ij} \geq 0 \qquad i = 1, \ldots, |M|,\ j = 1, \ldots, |N| \tag{5.6d}$$

Damit lässt sich nun das Modell als eine Verallgemeinerung des klassischen Transportproblems[2] interpretieren. Den Ausgangsorten A_i mit den Lagermengen a_i entsprechen die Ausführungsorte mit jeweils ein Auftrag m_i mit der Plandauer p_{m_i}. Den Bedarfsorten B_j entsprechen die Monteure n_j mit den Arbeitszeitkapazitäten b. Die Transportkosten c_{ij} sind wie in (5.4) definiert. Wird das Problem so modelliert, sind die Bedingungen, welche an das klassischen Transportproblem gestellt werden, nicht erfüllt. Zum Beispiel ist die Anzahl der Monteure nicht gleich der Summe der Plandauern, wie in (4.9) gefordert wird.

Gegenüber dem klassischen Transportproblem enthält die Nebenbedingung (5.6c) in der Summe zusätzlich einen Koeffizienten und auch die rechte Seite der Nebenbedingung (5.6b) ist modifiziert. Jedoch kann ein Zusammenhang zwischen all diesen Besonderheiten gefunden werden. Werden die Gleichungen (5.6b) mit p_{m_i} auf beiden Seiten multipliziert, ergibt sich:

$$\sum_{j=1}^{|N|} p_{m_i}x_{ij} = p_{m_i} \qquad i = 1, \ldots, |M|$$

Zudem wird eine neue Variable y_{ij} mit $y_{ij} = p_{m_i}x_{ij}$ eingeführt und im Modell wird x_{ij} durch $\frac{y_{ij}}{p_{m_i}}$ ersetzt. Damit ergibt sich das folgende Modell:

[2] Vgl. Kapitel 4.3.1

(AZ_{R2})

$$\text{ZF:} \qquad \sum_{i=1}^{|M|}\sum_{j=1}^{|N|} \frac{c_{ij}}{p_{m_i}}\, y_{ij} \rightarrow min \tag{5.7a}$$

$$\text{unter den NB:} \qquad \sum_{j=1}^{|N|} y_{ij} = p_{m_i} \qquad i = 1, \dots, |M| \tag{5.7b}$$

$$\sum_{i=1}^{|M|} y_{ij} = b \qquad j = 1, \dots, |N| \tag{5.7c}$$

$$y_{ij} \geq 0 \qquad i = 1, \dots, |M|,\ j = 1, \dots, |N| \tag{5.7d}$$

Dieses Modell erfüllt nun ebenso die Bedingung (4.9):

$$\sum_{i=1}^{|M|} p_{m_i} = |N| \cdot b \tag{5.8}$$

Aus dem verallgemeinerten Transportproblem (AZ_{R1}) ergibt sich somit das klassisches Transportproblem AZ_{R2} mit der Variable y_{ij}. Die Variable y_{ij} kann als Aufteilung der Arbeitszeit p_{m_i} auf den Monteur n_j interpretiert werden.

Nach der Ermittlung der optimalen Lösung Y^* muss diese noch zweilenweise durch p_{m_i} ($i = 1, \dots, M$) geteilt werden um die gesucht Lösung X^* zu erhalten. Durch die Zurückführung auf das klassische Transportproblem kann die Lösung mittels Vogelscher Approximation und anschließender Potentialmethode effizient ermittelt werden. Der Aufwand zur Lösung von (AZ_{R2}) ist geringer als die Lösung von (AZ_{R1}), da (AZ_{R2}) als verallgemeinertes Transportproblem nicht mit den oben genannten Methoden gelöst werden kann, sondern zum Beispiel mit dem Simplex Algorithmus. Die optimalen Lösungen beider Modelle sind wegen $x^*_{ij} = \frac{y^*_{ij}}{p_{m_i}}$ identisch.

Es sei $\widetilde{X} = \left(\widetilde{x}_{ij}\right)_{i=1,\dots,|M|}^{j=1,\dots,|N|}$ die optimale Lösung von (AZ_{R1}). Diese Lösung ist im Allgemeinen jedoch nicht zulässig für das Ausgangsproblem (AZ), da einige Werte $\widetilde{x}_{ij}$ nicht ganzzahlig sind, was der Aufteilung eines Auftrages auf mehrere Monteure entspricht.

Die optimale Lösung von (AZ_{R1}) hinsichtlich Beispiel 1 mittels eines linearen Algorithmus[3] ergibt ein Zielfunktionswert von $135,5$ km mit folgenden Lösungsmatrix:

$$\widetilde{X} = \left(\widetilde{x}_{ij}\right)_{i=1,\dots,10}^{j=1,\dots,4} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0,45 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0,55 \\ 0 & 0 & 0 & 0 & 0,167 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0,833 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}^T \tag{5.9}$$

und einer Arbeitszeit von $4,25\,h$ für jeden Monteur.

Die Lösung ist hinsichtlich (AZ_{R1}) optimale, jedoch für das Ausgangsproblem (AZ) nicht zulässig, da die Aufträge m_5 und m_{10} auf mehrere Monteuren aufgeteilt werden.

[3] Zur Ermittlung der Lösung wurde linprog() von MATLAB verwendet.

Definition 8 (Split)**.** *Gegeben ist die optimale Lösung $\widetilde{X}$ aus* (5.6)*. Wird ein Auftrag m_i mehreren Monteuren zugeordnet, d. h. es existieren für ein i mehrere j, für die $x_{ij} > 0$ gilt, wird ein solcher Auftrag als Split bezeichnet. Die Menge aller Splits sei*

$$M^s = \{i \in \mathbb{N} | \exists j : 0 < \widetilde{x}_{ij} < 1, \widetilde{x}_{ij} \in \widetilde{X}\}$$

Bei einem Split wurde der Auftrag zwar vollständig den Monteuren zugeordnet, jedoch ist diese Zuordnung nicht eindeutig. Das Verfahren, welches im Anschluss jedem Split genau einen Monteur zuweist, wird als *Splitauflösung* bezeichnet.

Die Lösung des Transportproblems lässt sich ebenso als bipartiter Graph[4] G interpretieren. Die erste Knotenmenge bilden die Aufträge M und die zweite Knotenmenge die Monteure N. Zwischen einem Knoten m_i aus M und einem Knoten n_j aus N existiert genau dann eine Kante, wenn $\widetilde{x}_{ij} > 0$ ist. Die Kantenbewertung sei der Wert von $\widetilde{x}_{ij}$.

Die optimale Lösung des relaxierten Problems zum Beispiel 1 wird in der Abbildung 5.2 mit Hilfe eines bipartiten Graphens dargestellt. Die linke Knotenmenge steht für die Monteure und die rechte Knotenmenge für die Aufträge. Die Kanten spiegeln die Zuordnungen wider. Bei den durchgezogen Linien ist der zugehörige Wert von $\widetilde{x}_{ij} = 1$ und bei den gestrichelten Linien liegt er zwischen $0 < \widetilde{x}_{ij} < 1$.

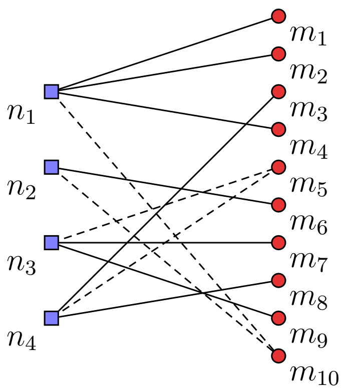


Abbildung 5.2: Darstellung der optimalen Lösung $\widetilde{X}$ des Beispiels 1 mit dem Modell (5.6) als bipartiter Graph

Eine wichtige Eigenschaft dieses bipartiten Graphens G ist, dass er keine Zyklen enthält, d. h. ausgehende von einem Knoten n_i ist es nicht möglich entlang der Kanten so zu gehen, dass der Knoten n_i wieder erreicht wird ohne eine Kante doppelt zu nutzen. Diese Eigenschaft begründet sich aus der linearen Unabhängigkeit der Spalten mit positiven Variablen x_{ij} einer Basislösung[5].

Die Splits beim Beispiel 1 sind nur die Aufträge m_5 und m_{10} und die Anzahl ist damit sehr klein. Im Allgemeinen lässt sich eine Aussage über die maximale Anzahl der Splits treffen. Der folgende Satz gibt eine obere Schranke an[6]:

Satz 2. *In einer optimalen Lösung von (AZ_R) ist die Anzahl der Splits kleiner als die Anzahl der Monteure $|N|$.*

[4]Ein bipartiter Graph ist ein einfacher Graph, dessen Knotenmenge sich in zwei disjunkte Teilmengen U und V zerlegen lässt, so dass die Knoten sowohl innerhalb U als auch innerhalb V nicht adjazent zueinander sind [10].

[5]Wären Zyklen im Graph vorhanden, dann würde sich eine lineare Abhängigkeit dieser Spalten ergeben [6].

[6]Satz und Beweis sind an [26] angelehnt und auf das gegebene Problem übertragen.

Zum Beweis des Satzes wird das folgende Lemma benötigt:

Lemma 1. *Ein zyklenfreier bipartiter Graph $G = (U \times V, E)$ besitzt maximal $|V| + |U| - 1$ Kanten.*

Dies hat die Ursache darin, das jeder zyklenfreie Graph auch als Baum bezeichnet wird und jeder Baum mit n Knoten besitzt maximal $n - 1$ Kanten [10]. Dies lässt sich auf den zyklenfreien bipartiten Graphen übertragen.

Beweis von Satz 2. Sei $G' = (M^s \times N, E^s)$ der bipartite Subgraph von G, welcher ausschließlich Kanten mit den Bewertungen $0 < \widetilde{x}_{ij} < 1$ der optimalen Lösung $\widetilde{X}$ besitzt, d. h. jede Kante repräsentiert einen Split. Da die Summe $\sum_{j=1}^{|N|} \widetilde{x}_{ij} = 1$ laut der ersten Nebenbedingung aus (5.6) ist, besitzt jeder Knoten aus M^s mindestens den Grad 2. Damit ist die Anzahl der Kanten mindestens doppelt so groß wie die Anzahl der Splits, d. h. $|E^s| \geq 2|M^s|$. Da G' zyklenfrei ist und damit das Lemma 1 Gültigkeit besitzt, folgt:

$$2|M^s| \leq |E^s| \leq |M^s| + |N| - 1 \Rightarrow |M^s| \leq |N| - 1 \qquad \square$$

Da die Anzahl der Monteure in der Praxis weit aus geringer ist, als die Anzahl der Aufträge, sind ebenso die Anzahl der Splits gering. Dies hat zur Folge, das bei der Splitauflösung nur ein relativ kleine Menge M^s betrachtet werden muss.

Bei der optimalen Lösung von (5.6) besitzt jeder Monteur die gleiche Summe b an Plandauern der zugeordneten Aufträge. Bei der Splitauflösung, kann dies nicht mehr gewährleistet werden. Es ergibt sich ein Ungleichgewicht in der Verteilung der Gesamtplandauer.

In der Literatur [26] wird vorgeschlagen bei der Splitauflösung einen Auftrag m_i nur solch einem Monteur n_j zuzuordnen, bei dem in der Lösung $\widetilde{X}$ die Variable $\widetilde{x}_{ij} > 0$ ist. Somit entsteht nur eine kleine Menge an Kombinationsmöglichkeiten, welche mit exakten Verfahren schnell zu lösen sind. Neben diesem Vorteil erscheint es auch heuristisch sinnvoll ausschließlich diese Monteure zu betrachten. Während der Lösung des allgemeinen Transportproblems fand eine Minimierung bezüglich der Gesamtentfernungen statt und die Summe der Plandauern war für jeden Monteur gleich, d. h. die Splits enthalten einen Kompromiss zwischen beiden Forderungen.

Die einfachste Möglichkeit die Splitauflösung vorzunehmen, ist die Zuordnung eines Splits m_i zu dem Monteur n_j mit dem größten Anteil bei $\widetilde{x}_{ij}$. Diese Zuordnung bewirkt, dass die Abweichung in der Gesamtplandauer von b gering ist, da die Lösung $\widetilde{X}$ bezüglich der Gesamtplandauer vollständig ausgeglichen ist. Jedoch finden die Distanzen c_{ij} bei dieser Art der Splitauflösung keine Beachtung. Sie ist somit unabhängig von der Zielfunktion, d. h. sie kann zu einer starken Vergrößerung des Zielfunktionswertes führen.

Beim Beispiel 1 ergab sich nach der Lösung von (AZ_R) die Lösungsmatrix $\widetilde{X}$ aus (5.9). Bei der Splitauflösung nach dem größten Anteil in $\widetilde{x}_{ij}$ wird Auftrag m_5 dem Monteur n_4 und Auftrag m_{10} dem Monteur n_2 zugewiesen. Es ergibt sich ein Zielfunktionswert von $141\,km$ und die Gesamtplandauern $[2h\,;\,6,5h\,;\,4h\,;\,4,5h]$ für die Monteure.

Liegt die Priorität jedoch in der Minimierung der Gesamtentfernungen, sollten bei der Splitauflösung diese mit berücksichtigt werden. Die eindeutige Zuordnung der Splits erfolgt in diesem Fall so, dass die Summe aller Entfernungen c_{ij} hinsichtlich der Aufträge aus M^s minimiert wird. Dabei wird die Forderung der Balance der Gesamtplandauer zunächst vernachlässigt. Es ergibt sich wiederum ein Optimierungsaufgabe, bei der ausschließlich die Splits betrachtet werden und die Nebenbedingung der gleichmäßigen Verteilung der Gesamtplandauer je Monteur entfällt.

Das Minimierungsproblem, welches nun zu lösen ist, lautet:

(S)

$$\text{ZF:} \qquad \sum_{i \in M^s} \sum_{j=1}^{|N|} \widehat{c}_{ij}\, x_{ij} \rightarrow min \tag{5.10a}$$

$$\text{unter den NB:} \qquad \sum_{j=1}^{|N|} x_{ij} = 1 \qquad i \in M^s \tag{5.10b}$$

$$x_{ij} \in \{0,1\} \qquad i \in M^s,\ j = 1, \ldots, |N| \tag{5.10c}$$

Dabei werden die Zielfunktionskoeffizienten c_{ij} so modifiziert, dass nur die Paare (i,j) mit $0 < \widetilde{x}_{ij} < 1$ bei der Optimierung betrachtet werden müssen, d. h.

$$\widehat{c}_{ij} = \begin{cases} c_{ij} & \text{, wenn gilt: } 0 < \widetilde{x}_{ij} < 1 \\ M_\infty & \text{, sonst} \end{cases} \tag{5.11}$$

mit M_∞ hinreichend groß.

Es handelt sich nun um ein binäres Minimierungsproblem mit einer geringen Anzahl an Variablen, so dass an dieser Stelle auf ein exaktes Verfahren zurückgegriffen werden kann. Es bietet sich an dieser Stelle ein Branch&Bound-Verfahren an.

Branch&Bound-Verfahren bei der Splitauflösung

Das Prinzip des Branch&Bound-Verfahrens wurde bereits im Kapitel 4.4 beschrieben. An dieser Stelle werden nun konkret die Verzweigungsregeln, Schranken ect. bei der Anwendung auf die Splitauflösung definiert.

Zunächst handelt es sich um ein Minimierungproblem und damit wird eine erste obere Schranke S_0 für den optimalen Zielfunktionswert von (5.10) benötigt. Diese kann zum Beispiel der Zielfunktionswert aus der oben beschriebenen Methode sein, bei der ein Split dem Monteur mit dem größten Anteil bei $\widetilde{x}_{ij}$ zugeordnet wird. Es ist zu beachten, dass der Zielfunktionswert von der optimalen Lösung $\widetilde{X}$ von (AZ_R) weder eine untere noch obere Schranke bildet, da die Bedingung der Gleichverteilung der Gesamtplandauer keine Beachtung mehr findet.

Zudem werden die Menge X_s^1 und X_s^0 benötigt. Diese beinhalten die Wahl der Variablen der einzelnen Teilprobleme. In der Wurzel des Lösungsbaums besteht X_s^1 aus allen Paare (i,j) für die $\widetilde{x}_{ij} = 1$ gilt. Diese entsprechen den eindeutigen Zuordnungen der Aufträge zu den Monteuren in der Lösung $\widetilde{X}$ und wird auf die Lösung X übertragen. Wird in einem Teilproblem $x_{kl} = 1$ gesetzt, so wird das Paar (k,l) zur Menge X_s^1 hinzugefügt. Die Menge X_s^0 ist zunächst leer und wird um das Paar (k,l) erweitert, wenn in einem Teilproblem $x_{kl} = 0$ gesetzt wird.

Die Wahl eines Splits aus der Menge M^s zum Verzweigen kann zufällig geschehen. In diesem Fall soll dies nach der Reihenfolge der Nummerierung der Aufträge und Monteure geschehen. Sei x_{kl} nun der Ausgangspunkt für die Verzweigung. Es entstehen zwei neue Teilprobleme P_1 mit $x_{kl} = 1$ und P_2 mit $x_{kl} = 0$. Im Teilproblem P_1 kann der Split k aus der Menge M^s entfernt und das Paar (k, l) zur Menge X_s^1 hinzugefügt werden. Zudem werden in P_1 alle weiteren Variablen $x_{kj} = 0$ für $j \neq l$ gesetzt. Beim Teilproblem P_2 ergibt sich die Menge X_s^0 zu $X_s^0 = \{(k, l)\}$.

Zur Berechnung der unteren Schranke eines Teilproblems, kann als Relaxation die zugehörige lineare Optimierungsaufgabe zu (S) verwendet werden:

(S_R)

$$\text{ZF:} \qquad \widehat{z} = \sum_{i \in M^s} \sum_{j=1}^{|N|} \widehat{c}_{i,j}\, x_{ij} \rightarrow \min \tag{5.12a}$$

$$\text{unter den NB:} \qquad \sum_{j=1}^{|N|} x_{ij} = 1 \qquad i \in M^s \tag{5.12b}$$

$$\sum_{(i,j) \in X_s^0} x_{ij} = 0 \tag{5.12c}$$

$$x_{ij} \in [0, 1] \qquad i \in M^s,\ j = 1, \ldots, |N| \tag{5.12d}$$

Diese wird für den Knoten P_1 bzw. P_2 gelöst und der Zielfunktionswert $\bar{z} = \widehat{z} + \sum_{(i,j) \in X_s^1} c_{ij}$ mit der obere Schranke S_0 verglichen. Ist dieser größer als S_0 kann der Knoten wegen Unzweckmäßigkeit gestrichen werden.

Unzulässigkeit kann nur in einem rechten Knoten mit $x_{kl} = 0$ eintreten. Dieser Knoten wird gestrichen, wenn kein weiteres k mit $\widetilde{x}_{ki} > 0\, \forall i$ existiert. In diesem Fall würde der Auftrag n_k keinem Monteur zugeordnet werden.

Die Verzweigung wird nun in einem zulässigen Knoten mit Wahl einer neuen Variable analog wie bei der ersten Verzweigung durchgeführt. Sind keine Splits mehr vorhanden, also gilt $M^s = \emptyset$, und alle Aufträge eindeutig den Monteuren zugeordnet, wurde im Lösungsbaum ein Blatt erreicht. Der zugehörige Zielfunktionswert bildet eine bessere obere Schranke S_1 für z^*, wenn der Wert kleiner als S_0 ist.

Beendet werden kann das Verfahren, wenn keine Teilprobleme mehr untersucht werden müssen. Das Blatt mit dem kleinsten Zielfunktionswert bildet die optimale Lösung der Splitauflösung. Es ist zu beachten, das die Entfernungen minimiert wurden und keinerlei Rücksicht auf das Gleichgewicht der Gesamtplandauer bei den einzelnen Monteuren genommen wurde.

Für das Beispiel 1 ergibt die Anwendung des beschriebenen Branch&Bound-Verfahrens nach der MLB-Regel den Lösungsbaum in der Abbildung 5.3. Als obere Schranke wird $S_0 = 141$, der Zielfunktionswert der Lösung durch die Splitauflösung nach dem größten Anteil, gewählt. In der Wurzel ist die Menge $X_s^1 = \{(1,1); (2,1); (3,4); (4,1); (6,2); (7,3); (8,4); (9,2)\}$ und $X_s^0 = \emptyset$. Ausgehend von der Menge $M^s = \{5, 10\}$ wird bei der ersten Verzweigung $x_{53} = 1$ und $x_{53} = 0$ gewählt. Im Knoten P_1 wird zu X_s^1 das Element $(5, 3)$ hinzugenommen. Der Zielfunktionswert $\bar{z}$ ergibt sich zu 125. Im Knoten P_2 wird der Auftrag m_3 nicht dem Monteur n_5 zugeordnet. Es ergibt sich X_S^0 zu $X_S^0 := \{(5, 3)\}$. Der Zielfunktionswert des zugehörigen Modells S mit $c_{53} = M_\infty$

berechnet sich zu $\hat{z} = 31$ und damit ist $\bar{z}_2 = 131$. Da der Zielfunktionswert $\bar{z}_1 < \bar{z}_2$ wird der Knoten P_1 zur weiteren Verzweigung verwendet. Das einzige Element, welches in M^s enthalten ist, ist der Auftrag m_{10}. Unterschieden wird in $x_{101} = 1$ und $x_{101} = 0$. Der Knoten P_3 mit x_{101} besitzt den Zielfunktionswert $\bar{z}_3 = 125$. Da dieser Knoten ein Blatt ist und $\bar{z}_3 < S_0$ ist, wird die neue obere Schranke $S_1 = \bar{z}_3 = 125$ gesetzt. Damit können die Knoten P_4 und P_2 jeweils wegen Unzweckmäßigkeit gestrichen werden.

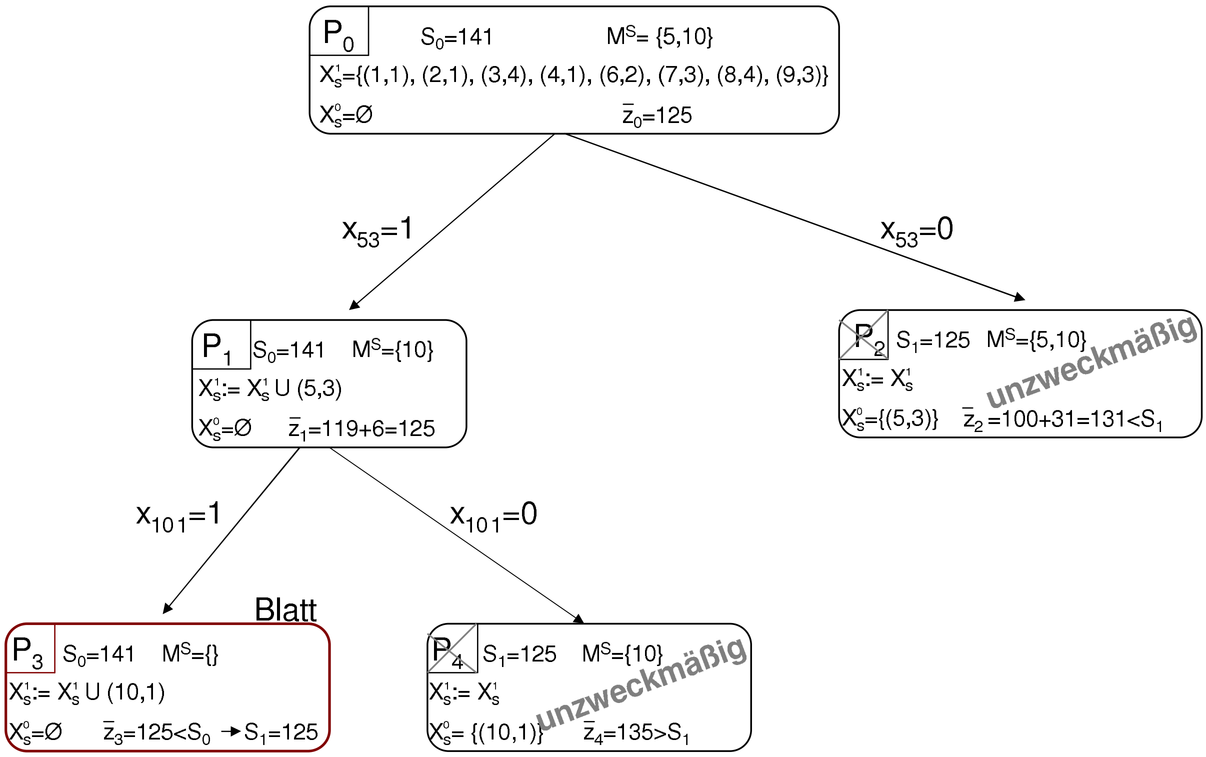


Abbildung 5.3: Lösungsbaum der Anwendung des Branch&Bound-Verfahrens auf das Beispiel 1

Für die optimale Lösung nach dem Branch&Bound-Verfahren ergibt sich die Zuweisung $m_5 \rightarrow n_3$ und $m_{10} \rightarrow n_1$. Die Lösung für die Auftragszuordnung lässt sich aus der Menge $X_s = \{(1,1),(2,1),(3,4),(4,1),(5,3),(6,2),(7,4),(8,3),(9,3),(10,1)\}$ ablesen. Der Zielfunktionswert beträgt $117\,km$ und die Gesamtplandauern der Monteure ergibt sich zu $[7h; 1,5h; 4h; 3h]$.

Es wird ersichtlich, dass keine gleichmäßige Verteilung der Plandauern nach der Splitauflösung mehr vorliegt. Eine Ursache für die schlechte Verteilung ist die Wahl des zu kleinen Beispiels. Dieses Beispiel hat nur wenige Aufträge im Bezug auf den Monteure und große Abweichungen der Plandauern zwischen den Aufträgen. Bei größerer Beispiel wird die Splitauflösung sich nicht so stark auf das Gleichgewicht der Verteilung der Plandauern auswirken.

Weitere Möglichkeiten der Splitauflösung

Es wurden bereits zwei extreme Möglichkeiten der Splitauflösung vorgestellt. Die erste geschieht nur unter der Berücksichtigung des Gleichgewichts der Verteilung der Gesamtplandauern und die zweite nur unter Berücksichtigung der Entfernungen. Eine weitere Möglichkeit ist die Bildung eines Kompromisses zwischen beiden, also die Berücksichtigung der Gesamtplandauer als auch die der Entfernungen. Dazu wird die Zielfunktion in (S) modifiziert. Da beide Forderungen voneinander unabhängig sind, können sie linear verknüpft werden. Die Entfernungen werden dazu normiert und das Gleichgewicht der Gesamtdauer wird über die mittlere quadratische Abweichung gemessen.

$$\text{ZF:} \qquad \alpha \sum_{i \in M^s} \sum_{j=1}^{|N|} c'_{ij} x_{ij} + \beta \sum_{j=1}^{|N|} (b - \sum_{i \in M^s} p'_{m_i} x_{ij})^2 \rightarrow \min$$

$$\text{mit:} \qquad c'_{ij} = \begin{cases} \frac{d(m_i, n_j)}{d_{max}} & \text{, wenn gilt: } 0 < \widetilde{x}_{ij} < 1 \\ M_\infty & \text{, sonst} \end{cases}$$

$$p'_{m_i} = \begin{cases} p_{m_i} & \text{, wenn gilt: } 0 < \widetilde{x}_{ij} < 1 \\ M_\infty & \text{, sonst} \end{cases}$$

Dabei ist d_{max} die maximal Entfernung in der Entfernungsmatrix und M_∞ hinreichend groß. Mit Hilfe der Parameter $\alpha \in [0,1]$ und $\beta \in [0,1]$ mit $\alpha + \beta = 1$ kann der Einfluss der beiden Forderungen gesteuert werden. Die Parameter werden nach der Priorität der Forderungen gewählt.

Das modifizierte Optimierungsproblem kann ebenso mittels Branch&Bound-Verfahren, wie oben beschrieben, gelöst werden. Für die geeignete Wahl von α und β bedarf es einiger Erfahrung, welche durch verschiedene Test erlangt werden können.

5.2.4 Zusammenfassung

Die Formulierung als verallgemeinertes Zuordnungsproblem liefert mit heuristischen Verfahren eine gute Näherungslösung. Jedoch beeinflusst die Wahl des Parameters δ den Zielfunktionswert und die Verteilung der Gesamtplandauern unter den Monteuren. Wird der Parameter groß gewählt, so ergibt sich eine geringe Gesamtentfernung, jedoch eine schlechte Verteilung der Gesamtplandauer. Wird der Parameter kleiner gewählt, wird eine nahezu gleichmäßige Verteilung der Gesamtplandauer erreicht, jedoch auf Kosten des Zielfunktionswertes. Das Verfahren eignet sich vor allem für größere Problemstellungen, da dann die Sensibilität der Forderungen durch die Wahl von δ nicht so groß ist.

Mit Hilfe des Parameters δ ist eine Steuerung der Gewichtung beider Forderung möglich. Wie stark dieser Einfluss ist und die Schlussfolgerung wie der Parameter δ somit gewählt werden sollte, muss in einem ausführlichen Test mit praktischen Daten analysiert werden. Dies würde den Rahmen dieser Masterarbeit, in der zunächst nur mögliche Verfahren aufgezeigt werden, jedoch sprengen und könnte Aufgabe einer nächsten Arbeit sein.

Auch bei der Lösung mittels Relaxation, kann die Forderung AZ(b), dass alle Monteure eine nahezu gleiche Gesamtplandauer der zugeordneten Aufträge besitzen sollen, bei kleinen Problemstellung nur schwer erfüllt werden. Ist die Anzahl der Aufträge jedoch weitaus höher als die Anzahl der Monteure, entstehen prozentual nur kleine Abweichungen zwischen den einzelnen Gesamtplandauern.

Sowohl die Vogelsche Approximation mit anschließender Potentialmethode, angewendet auf das klassische Transportproblem, als auch das Branch&Bound-Verfahren, angewendet auf die Splitauflösung liefern optimale Lösungen für das jeweilige Problem. Jedoch wird die zweite Nebenbedingung des Ausgangsproblems (5.3) aufgelockert. Eine gleiche Verteilung der Gesamtplandauern auf die Monteure wird nicht mehr gefordert. Somit ist die Lösung X, welche sich nach der Splitauflösung ergibt, für das Ausgangsproblem nicht zulässig.

Wie bereits erwähnt existiert im Allgemeinen keine zulässige Lösung für das Ausgangsproblem (5.3) und somit auch keine optimale Lösung. Sowohl die Lösung mittels verallgemeinerten Zuordnungsproblem als auch die Lösung mittels Relaxation liefern fast zulässige Lösungen, welche hinsichtlich der Gesamtentfernungen minimiert wurden. Nicht zulässig bedeutet, dass die Lösung nicht alle Nebenbedingungen erfüllt, also das nicht alle Monteure die ein und die selbe Gesamtplandauer besitzen. Jedoch sind die Algorithmen so konstruiert, dass die Gesamtplandauer wenig voneinander abweichen. Für das praktische Problem von envia NSG ist die Lösung der Auftragszuordnung anwendbar.

Welches Verfahren bevorzugt werden sollte, ist an dieser Stelle schwierig herauszustellen. Für beide Ansätze existieren bereits schnelle Lösungsverfahren. Zwar muss bei der ersten Möglichkeit nur das verallgemeinerte Zuordnungsproblem gelöst werden und bei der zweiten Möglichkeit zunächst das Transportproblem und anschließend die Splitauflösung, d. h. es müssen zwei Lösungsalgorithmen programmiert und nacheinander ausgeführt werden, doch wird zu Lösung des verallgemeinerten Zuordnungsproblem ebenso ein Eröffnungsverfahren und anschließend ein Verbesserungsverfahren benötigt. Damit ist vor allem die Implementation der Algorithmen ausschlaggebend für die Laufzeit der Verfahren.

Da kein Optimum existiert und in beiden Verfahren Möglichkeiten bestehen um die Forderungen AZ(a) und AZ(b) zu gewichten, ist auch in diesem Punkt keines zu bevorzugen. Detaillierte Untersuchungen mit realen Datenmaterial würden die Vor- und Nachteile beider Verfahren deutlicher herauskristallisieren und somit könnte eine Wahl getroffen werden. Diese Untersuchungen könnte der Grundstein einer weiteren Arbeit sein. An dieser Stelle bleibt es dem Leser überlassen, welches Verfahren er bevorzugt.

Ob nun das Lösungsverfahren mit Anpassung der rechten Seite oder mit Relaxation gewählt wird, als Ergebnis wird die Lösungsmatrix X erhalten. Aus der Spalte j von X kann dann abgelesen werden, welche Aufträge dem Monteur j zugewiesen wurde. Das Resultat der Auftragszuordnung sind die Mengen M_j der zugeordneten Aufträge zum Monteur j.

5.3 Tagestouren

Während der Auftragszuordnung wurde die Gesamtmenge der Aufträge auf die Monteure gleichmäßig aufgeteilt (siehe Abschnitt 5.2). Als Ergebnis erhält jeder einzelne Monteur eine Menge an Aufträgen M_j. Das Ziel in diesem Abschnitt ist, einen Algorithmus zu entwickeln, welcher für jeden Monteur die einzelnen Tagestouren festlegt. Dabei werden aus der Menge der zugeordneten Aufträge eines Monteurs, die Aufträge für einzelne Touren heraus gesucht und die Reihenfolge festgelegt. Zunächst soll dies nur unter der Forderung des betrieblichen Ziels BZ 3, der Minimierung der Fahrzeit, geschehen. Alle weiteren betrieblichen Ziele, welche bei den Tagestouren zudem verfolgt werden können, werden in diesem Abschnitt nach und nach diskutiert.

5.3.1 Modellierung des Grundproblems

Die Tagestouren beinhalten vom Grunde her ein in der Praxis weit verbreitetes mathematisches Problem: die Tourenplanung oder auch Vehicle Routing Problem (VCR) genannt. Jedoch sind an jedem praktischen Problem immer Besonderheiten gebunden, welche Veränderung des Modells fordern und somit auch eine Anpassung der Lösungsverfahren. So wird in diesem Abschnitt zunächst von einem stark vereinfachten Problem ausgehend ein Modell mit Lösungsvorschlägen angegeben und im Anschluss Schritt für Schritt die weiteren Ziele und Spezialfälle, wie z. B. Mehrtageseinsätze hinzugenommen. Als Grundlage wird die Theorie aus dem Abschnitt 4.3.4 vorausgesetzt.

Im Folgenden wird das Tagestourenproblem nur für einen Monteur betrachtet, analog ergibt sich dies für alle weiteren Monteure. Es sei $V = \{v_1, v_2, \ldots, v_{|V|}\}$ die Mengen der Hauptvorgänge, welche sich aus der zugeordneten Auftragsmenge ableitet. Dabei ist v_i der zugehörige Hauptvorgang zum i.ten Auftrag der Auftragsmenge M_j, welche bei der Auftragszuordnung dem Monteur zugeordnet wurde. Zunächst wird nur der Hauptvorgang mit Plandauer und Ort der Ausführung betrachtet. Die einzelnen Hauptvorgänge seien unabhängig voneinander, d. h. eine Reihenfolge der Abarbeitung ist nicht gegeben. Zudem werden anfangs der Ausführungszeitraum und die Priorität der Vorgänge vernachlässigt.

Das Ziel ist zunächst die Bildung der Tagestouren, d. h. die Bestimmung einer Tour für jeden Tag und jeden Monteur mit minimaler Gesamtlänge. Dabei fährt der Monteure jeden Tag vom selben Standort los. Dieser Standort wird zukünftig mit *Depot* bezeichnet und mit v_0 gekennzeichnet.

Zudem sei die Entfernung $d(v_i, v_j)$ zwischen den Ausführungsorten aller einzelnen Vorgänge sowie zwischen allen Ausführungsorten und dem Depot in einer Entfernungsmatrix gegeben. Anders als bei der Auftragszuordnung wird nun die direkte Routenentfernung abgespeichert, diese kann zum Beispiel mit Hilfe von routenfähigem Material bestimmt werden. Wie bereits im Kapitel 3 beim Optimierungsziel V2, wird die Fahrzeit in linearer Abhängigkeit zu Fahrstrecke angenommen. Statt der Minimierung der Gesamtfahrstrecke wird die Minimierung der Gesamtfahrzeit vorgenommen.

Die Fahrzeitenmatrix entspricht der Kostenmatrix C des Tagestourenproblems für den Monteur und ergibt sich durch Division der Entfernungsmatrix durch eine Durchschnittsgeschwindigkeit $\bar{\mathrm{v}}$. Da zudem die Plandauer eines Vorganges konstant ist, d. h. unabhängig von der Fahrzeit ist, kann diese in die Fahrzeitenmatrix einbezogen werden. Dabei wird analog zu (4.20) zur Fahrzeit zwischen zwei Orten immer die Hälfte der Plandauer des Ausgangsortes und die Hälfte der Plandauer des Zielortes addiert. Die Plandauer im Depot wird gleich Null gesetzt $(p_0 = 0)$. Eine Fahrt vom Ausführungsort des Vorgang v_i zum selben Ort ist nicht sinnvoll, was mathematisch durch eine hinreichend große Zahl M_∞ realisiert wird. Die Einträge der Fahrzeitenmatrix ergeben sich somit zu:

$$c_{ij} := \begin{cases} \frac{p_i}{2} + \frac{1}{\mathrm{v}} \cdot d(v_i, v_j) + \frac{p_j}{2} & \text{, für } i \neq j \\ M_\infty & \text{, für } i = j. \end{cases} \tag{5.13}$$

Das Problem der Tagestouren ist ein längenbeschränktes Tourenproblem, wie bereits das Beispiel VRP aus dem Abschnitt 4.3.4. Die Knotenmenge N entspricht der Menge der Ausführungsorte der Vorgänge V mit dem Depot v_0. Die Anzahl der Touren sind in diesem Fall die Anzahl der Tage, welche der Monteur benötigt um alle Vorgänge der Menge V abzuarbeiten. Zunächst sei diese Anzahl unbekannt bzw. es bestehe keine Einschränkung.

Eine mögliche Tagestour T_k ist eine Teilmenge der Vorgangsmenge V und beinhaltet die Reihenfolge der Abarbeitung der Vorgänge. Jeder Monteur verfügt über eine obere Schranke A an Arbeitszeit, d. h. die Summe an Plandauer der Vorgänge und Fahrzeiten einer Tour darf diesen Wert A nicht überschreiten. Innerhalb dieser Arbeit wird die maximale Arbeitszeit pro Tag auf $A = 9h$ festgesetzt[7].

Das Tagestourenproblem wird als binäres lineares Modell angegeben. Dabei werden wie bereits im Abschnitt 4.3.4 die Variablen x_{ijk} und y_{ik} benötigt. Diese sind wie folgt definiert:

$$x_{ijk} = \begin{cases} 1, & \text{wenn in der Tour } T_k \text{ vom Vorgang } v_i \text{ zu } v_j \text{ gefahren wird.} \\ 0, & \text{sonst} \end{cases} \tag{5.14}$$

$$y_{ik} = \begin{cases} 1, & \text{wenn der Vorgang } v_i \text{ in der Tour } T_k \text{ enthalten oder } i = 0 \text{ ist.} \\ 0, & \text{sonst.} \end{cases} \tag{5.15}$$

Um zu sichern, dass in jeder Tour das Depot enthalten ist, wird $y_{0k} = 1 \, \forall k$ gesetzt. Für $i = 0$ ist y_{ik} somit keine Optimierungsvariable, sondern festgelegt.

Das Modell (4.27) lässt sich nun auf das Tagestourenproblem übertragen:

[7] Dieser Wert ist eine Annahme und muss in der Praxis durch Tests bestimmt werden.

(TT_1)

$$\text{ZF:} \quad \sum_{k=1}^{m} \sum_{i=0}^{|V|} \sum_{j=0}^{|V|} c_{ij}\, x_{ijk} \rightarrow min \tag{5.16a}$$

$$\text{unter den NB:} \quad \sum_{i=0}^{|V|} \sum_{j=0}^{|V|} c_{ij} x_{ijk} \leq A \qquad \forall k = 1, \ldots, m \tag{5.16b}$$

$$\sum_{k=1}^{m} y_{ik} = 1 \qquad \forall i = 1, \ldots, |V| \tag{5.16c}$$

$$\sum_{i=0}^{|V|} x_{ijk} \leq y_{ik} \qquad \forall k = 1, \ldots, m;\ j = 0, \ldots, |V| \tag{5.16d}$$

$$\sum_{j=0}^{|V|} x_{ijk} \leq y_{jk} \qquad \forall k = 1, \ldots, m;\ i = 0, \ldots, |V| \tag{5.16e}$$

$$\sum_{(i,j) \in S} x_{ijk} \leq |S| - 1 \qquad S \subset V \cup v_0, |S| \geq 1, k = 1, \ldots, m \tag{5.16f}$$

$$x_{ijk} \in \{0, 1\} \qquad \forall\, i, j = 0, \ldots, |V|;\ k = 1, \ldots, m \tag{5.16g}$$

$$y_{0,k} = 1 \wedge y_{ik} \in \{0, 1\} \qquad \forall\ i = 1, \ldots, |V|;\ k = 1, \ldots, m \tag{5.16h}$$

Die Zielfunktion minimiert die Gesamtfahrzeit über alle Touren. Die erste Nebenbedingung gewährleistet, dass die maximale Arbeitszeit in jeder einzelnen Tour nicht überschritten wird. Damit jeder Vorgang in genau einer Tour enthalten ist, wird die Nebenbedingung (5.16c) aufgestellt. Die Nebenbedingungen (5.16d) und (5.16e) stellen sicher, dass zu jedem Vorgang genau einmal hingefahren und genau einmal wieder weggefahren wird. Zur Vermeidung von Kurzzyklen wird analog wie in (4.17) die Nebenbedingung (5.16f) formuliert.

Das Savings-Verfahren und das 2 bzw. 3-opt-Verfahren als Verbesserungsverfahren können auf das Problem (TT_1), wie in Abschnitt 4.4.6 und 4.4.7 erläutert, angewendet werden. Das Savings-Verfahren liefert m Tagestouren, wobei m möglichst minimal ist. Die Touren können den m Tagen beliebig zugeordnet werden, da bisher keine weiteren zeitlichen Bedingungen oder Abhängigkeiten zwischen den Vorgänge gegeben sind. Mit Hilfe des 2 bzw. 3-opt-Verfahrens, kann die Lösung im Anschluss verbessert werden.

5.3.2 Erweiterung: Abhängigkeit zwischen den Vorgängen

Bisher wurden nur die Hauptvorgänge betrachtet. Jedoch ist bei envia NSG ein Auftrag in einen Hauptvorgang und verschiedene Nebenvorgänge unterteilt (siehe Kapitel 2). Der Hauptvorgang besitzt im Allgemeinen den größten Anteil an der Plandauer und ist damit entscheidend bei der Tagesarbeitszeit. Jedoch sind die Nebenvorgänge mit eventuell sehr kleiner Plandauer ebenso mit einzuplanen. Die Nebenvorgänge können in drei Klassen eingeteilt werden:

Klasse 1 Nebenvorgänge, welche am selben Tag wie der Hauptvorgang und vom eigenen Personal ausgeführt werden.

Klasse 2 Nebenvorgänge, welche an einem anderen Tag als der Hauptvorgang mit eigenen Personal ausgeführt werden.

Klasse 3 Nebenvorgänge, welche mit Fremdpersonal oder nicht direkt vom Monteur selbst ausgeführt werden.

Bei den Vorgängen der ersten Klasse ändert sich am oben beschriebenen Vorgehen nichts. Da diese Vorgänge immer direkt vor bzw. nach dem Hauptvorgang abzuarbeiten sind. Bei den Tagestouren werden diese mit dem Hauptvorgang *verschmolzen*, d. h. nicht einzeln betrachtet, sondern zum Hauptvorgang hinzugezogen und die Plandauer der Nebenvorgänge zur Plandauer p_j des Hauptvorganges addiert. Ausnahmen bilden Aufträge deren Hauptvorgang von einer Fremdfirma übernommen werden und deren Nebenvorgänge (z. B. Schaltungen) von eigenen Personal bearbeitet werden. Dieser Sonderfall wird später extra diskutiert (siehe 5.3.4).

Die Vorgänge der dritten Klasse sind ebenso unproblematisch, da diese nicht geplant werden müssen. Vorgänge mit der Kennzeichnung *Fremdleistung*, werden an andere Firmen abgegeben und müssen bei den Tagestouren für einen Monteur nicht beachtet werden. Dies gilt ebenso für Vorgänge, welche von anderem Personal als dem Monteur selbst zu erledigen sind. Hierunter zählen zum Beispiel die Schaltanträge.

Größere Schwierigkeiten bringen die Vorgänge der Klasse 2. Durch diese entstehen Abhängigkeiten zwischen den einzelnen Vorgängen, d. h. nicht alle Vorgänge können in einer beliebigen Reihenfolge abgearbeitet werden. Ein typisches Beispiel dafür ist die Verteilung von Abschaltinformationen. Diese müssen im Allgemeinen $2 - 3$ Tage vorher an die betreffenden Kunden ausgeteilt werden. Somit muss solch ein Nebenvorgang innerhalb dieser Relativzeit zum Hauptvorgang eingeplant werden. Zu jedem Hauptvorgang können mehrere Nebenvorgänge der Klasse 2 gehören.

Da Nebenvorgänge der Klasse 1 und 3 im Weiteren nicht mehr betrachtet werden müssen, sind im Folgenden immer Nebenvorgänge der Klasse 2 gemeint, ohne das extra darauf hingewiesen wird.

Zum Modell (5.16) ergeben sich somit einige Änderungen. Die Menge alle Nebenvorgänge sei NV. Sie kann mit der Vorgangsmenge V zu V' vereinigt werden, da die Nebenvorgänge auch als separate Vorgänge betrachtet werden können ($V' = V \cup NV$). Die Nummerierung der Nebenvorgänge schließt sich der Nummerierung der Hauptvorgänge an ($NV = \{v_{|V|+1}, v_{|V|+2}, \ldots, v_{|V'|}\}$). Mit Hilfe der Indexmenge $I = \{i_h\}_{h=|V|+1,\ldots,|V'|}$ wird die Zugehörigkeit der Nebenvorgänge zum Hauptvorgang abgespeichert. Dabei gilt $i_h = l$, falls der h-te Vorgang $(v_h \in NV)$ Nebenvorgang zum Hauptvorgang $v_l \in V$ ist.

Das Intervall $[t_h^u, t_h^o]$ in Tagen gibt das Zeitfenster für den Nebenvorgang $v_h \in NV$ an, in dem dieser relativ zum Hauptvorgang i_h eingeplant werden muss. Negative Werte des Intervalls, besagen, dass der Nebenvorgang vor und positive Werte, dass dieser nach dem Hauptvorgang abgearbeitet werden muss.

Die Arbeitszeitmatrix C' ergibt sich aus der Gleichung (5.13) mit der erweiterten Menge V'. Zum Modell (TT_1) wird nun die folgende Nebenbedingung hinzugefügt:

$$y_{i_h k} - \sum_{s \in [t_h^u, t_h^o] \cap Z} y_{hk+s} = 0 \qquad \forall i_h \in I \tag{5.17}$$

Diese garantiert, dass die abhängigen Vorgänge in der richtigen Reihenfolge und innerhalb der Relativzeit eingeplant werden.

Zum besseren Verständnis ein kleines Beispiel: Es gelte, dass die k.te Tour dem k.ten Planungstag entspricht. Gegeben seien 4 Hauptvorgänge $V = \{v_1, v_2, v_3, v_4\}$. Die zugehörigen Nebenvorgänge mit dem Intervall der Relativzeiten sind in der folgenden Tabelle aufgezeigt.

Hauptvorgang	Nebenvorgang	Intervall der Relativzeit
v_1	-	
v_2	v_5	[-5, -4]
v_3	v_6	[2, 2]
v_4	v_7, v_8	[-5, -4], [3, 4]

Die Menge der Nebenvorgänge ergibt sich zu $NV = \{v_5, v_6, v_7, v_8\}$ mit der Indexmenge $I = \{2, 3, 4, 4\}$.

Zum ersten Vorgang gehört kein Nebenvorgang, so dass dieser ohne weitere Einschränkungen eingeordnet werden kann. Zum zweiten Hauptvorgang gehört der Nebenvorgang v_5. Dieser besitzt ein negatives Intervall der Relativzeiten und muss dem zufolge vorher eingeplant werden. Ein typisches Beispiel dafür ist die Verteilung von Abschaltinformationen. Wird v_2 nun in der 10. Tour eingeplant ($k = 10$), dann muss der Vorgang v_5 der 5. oder 6. Tour zugeordnet werden. Die Gleichung (5.17) ergibt sich für diesen Fall zu:

$$y_{2\,10} - \sum_{s \in [-5,-4] \cap Z} y_{5\,10+s} = 0$$

$$y_{2\,10} - (y_{5\,5} + y_{5\,6}) = 0$$

Da $y_{2\,10}$=1 ist, muss für den Nebenvorgang v_5 entweder $y_{5\,5} = 1$ oder $y_{5\,6} = 1$ gelten.

Analog ergibt sich dies für den Hauptvorgang v_3:

$$y_{3\,k} - \sum_{s \in [2,2] \cap Z} y_{6\,k+s} = 0$$

$$y_{3\,k} = y_{6\,k+2}$$

Hier ist der Nebenvorgang v_6 jedoch genau 2 Tage nach dem Hauptvorgang einzuordnen, d. h. bei $y_{3\,k} = 1$ muss $y_{6\,k+2} = 1$ gelten. Der letzten Hauptvorgang besitzt zwei Nebenvorgänge. Der erste Nebenvorgang v_7 muss 4 bis 5 Tage vorher eingeplant werden und der zweite Nebenvorgang v_8, z. B. eine Kontrolle, soll 3 bis 4 Tage im Nachhinein erledigt werden.

Neben dem Modell müssen nun auch die Lösungsverfahren angepasst werden. Beim Savings-Verfahren dürfen zwei Vorgänge v_h, v_l, bei dem einer Nebenvorgang zu anderen ist, d. h. $i_h = l$ oder $i_l = h$, nicht der selben Tour zugeordnet werden. Dies muss nun neben der Einhaltung der Arbeitszeitgrenze geprüft werden. Im Anschluss kann die Zuweisung der Touren zu den Tagen nicht beliebig geschehen, sondern die Touren mit einem Nebenvorgang müssen innerhalb des Relativzeitintervalls zu Touren mit dem zugehörigen Hauptvorgang eingeordnet werden, d. h. bei der Festlegung der Reihenfolge muss die Bedingung (5.17) erfüllt sein. Jedoch kann der Fall auftreten, dass somit keine zulässige Zuordnung der Touren zu den Tagen mehr möglich ist, da sich die Relativzeiten der Nebenvorgänge mit der Zuordnung der Hauptvorgänge zu den Touren widersprechen.

Dies wird an einem Beispiel deutlich. Gegeben sei ein Auszug aus einer Zuordnung von Vorgängen zu Touren durch das Savings-Verfahren in der folgenden Tabelle.

Vorgang	zugehöriger Nebenvorgang mit Relativzeit	zugeordnete Tour
v_1	v_{11} mit [-3,-2]	T_k
v_2	v_{12} mit [-7,-6]	T_l
$\vdots$	$\vdots$	$\vdots$
v_{11}	-	T_l
v_{12}	-	T_k

Im Anschluss zu der Zuordnung soll nun die Reihenfolge aller Touren festgelegt werden. Wenn die Tour T_k zum Beispiel den 8. Tag zugeordnet wird, müsste die Tour T_l aufgrund der Relativzeit des Nebenvorgangs v_{11} dem 5. oder 6. Tag zugeordnet werden. Dies würde jedoch der Relativzeit des Nebenvorgangs $v_{12} \in T_k$ zum Hauptvorgang $v_2 \in T_l$ widersprechen. Bei Einhaltung der Relativzeit des Nebenvorgangs v_{12} zu v_2 und der Zuordnung der Tour T_k sechs bis sieben Tage vor der Tour T_l, würde dies dem Relativzeiten des Nebenvorgangs v_{11} widersprechen. Eine zulässige Festlegung der Touren zu Tagen ist somit nicht möglich.

Wie bereits erwähnt ist die Plandauer der Nebenvorgänge im Vergleich zu den Hauptvorgängen relativ gering. Eine weitere Möglichkeit ist, zunächst nur die Planung der Hauptvorgänge vollständig vorzunehmen und die Nebenvorgänge nicht zu beachten. Im Anschluss wird für jeden Hauptvorgang der zugehörige Nebenvorgang in eine Tagestour, die in dem erforderlichen Zeitfenster liegt, eingeplant. Dabei kann es jedoch zur Überschreitung der Arbeitszeitgrenze kommen. Um dies zu verhindern bzw. die Überschreitung so gering wie möglich zu halten, wird die obere Grenze A bereits vor der Zuordnung der Hauptvorgänge um einen Wert herabgesetzt.

Dazu wird die Summe der Plandauern aller Hauptvorgänge $S_V = \sum_{j:v_j \in V} p_j$ und Nebenvorgänge $S_{NV} = \sum_{i:v_i \in NV} p_i$ berechnet. Der zeitliche Anteil der Nebenvorgänge ergibt sich zu $S_{NV}/(S_V + S_{NV})$. Dies wird nun anteilsmäßig von der Arbeitszeitgrenze abgezogen:

$$A' = A - \frac{S_{NV}}{S_V + S_{NV}} \cdot A = A \cdot (1 - \frac{S_{NV}}{S_V + S_{NV}}) \tag{5.18}$$

Dabei wurden die Fahrzeiten bisher vernachlässigt. Diese kann zusätzlichen durch Abzug eines Durchschnittswert für die Fahrt zu einem Vorgang von A' berücksichtigt werden. Die Anpassung des Wertes kann durch einen Test mit realen Daten geschehen.

Nachdem auch die Nebenvorgängen den Touren zugeordnet wurden, kann eine Optimierung innerhalb der einzelnen Touren vorgenommen werden. Da die Anzahl der Vorgänge pro Tour gering ist, im Allgemeinen gilt $|T_k| < 10$ (siehe [34]), kann die optimale Rundreise für jede einzelne Tour, d. h. eine optimale Abfolge der Neben- und Hauptvorgänge eines jeden Tages, ermittelt werden. Somit wird die Fahrzeit und damit auch die Arbeitszeit der Tagestouren minimiert.

5.3.3 Erweiterung: Ausführungszeiträume

Jeder Auftrag besitzt einen Ausführungszeitraum. Dieser vererbt sich auf den Hauptvorgang. Er gibt die beiden Daten an, ab wann ein Vorgang frühstens und bis wann spätestes abgearbeitet werden soll. Im Folgenden sei dieser Zeitraum als Zeitintervall $[s_i, e_i]$ in Tagen im Bezug auf den aktuellen Tag der Planung gegeben. Ist der Wert für s_i negativ, so liegt das frühst möglichste Datum für die Abarbeitung in der Vergangenheit, d. h. der Vorgang kann ab den aktuellen Tag eingeplant werden. Ist jedoch ebenso $e_i \leq 0$, so liegt der Vorgang in der Vergangenheit und wurde nicht im vorgeschriebenen Zeitraum eingeplant. Dieser Hauptvorgang wird mit seinen möglichen

Nebenvorgängen aus der Menge V herausgenommen und nicht mehr betrachtet, sondern muss manuell vom Dispatcher eingeplant werden.

Zur Verdeutlichung der Zeitintervalle ein kleines Beispiel: Der Plantag x sei der 01.06.2010 und gegeben sind die vier Hauptvorgänge v_1, v_2, v_3, v_4. Die Ausführungszeiträume mit den zugehörigen Zeitintervallen sind in der folgenden Tabelle angegeben:

Vorgänge	Ausführungzeitraum	Zeitintervall
v_1	vom 01.05.2010 bis 30.06.2010	[-31, 29]
v_2	vom 15.05.2010 bis 20.05.2010	[-16,-11]
v_3	vom 15.06.2010 bis 15.07.2010	[14, 44]
v_4	Terminvorgang am 01.07.2010	[30, 30]

Veranschaulicht werden diese Ausführungszeiträume am Zeitstrahl in der Abbildung 5.4. Der Vorgang v_1 kann ab sofort eingeplant werden, da der aktuelle Tag innerhalb des Ausführungszeitraumes liegt und damit $s_1 < 0$ ist. Beim zweiten Vorgang ist keine automatische Einordnung mehr möglich, der Ausführungszeitraum liegt in der Vergangenheit. Der Ausführungszeitraum des dritten Vorgangs v_3 liegt in der Zukunft. Dieser kann frühsten in 14 Tagen eingeordnet werden. Der vierte Vorgang ist ein Terminvorgang[8]. Bei Terminvorgängen gilt $s_i = e_i$. Der Vorgang v_4 ist genau in 30 Tagen einzuordnen.

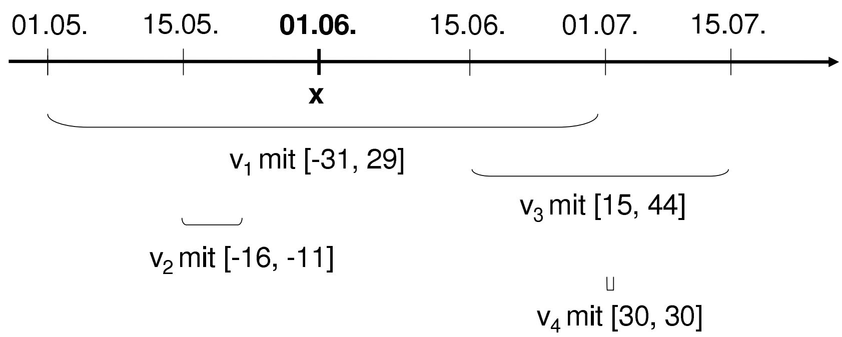


Abbildung 5.4: Vier Vorgänge mit den zugehörigen Ausführungszeiträumen

Das Zeitintervall $[s_i, e_i]$ wird dem Hauptvorgang zugeordnet und ist nicht mit den Relativzeiten der Nebenvorgänge zu verwechseln. Mit der zeitliche Bindung aller Vorgänge wird die Nebenbedingung (5.16c) im Modell (TT_1) modifiziert. Die zweite Nebenbedingung ergibt sich nun zu:

$$\sum_{k=\max\{0,s_i\}}^{\min\{e_i,m\}} y_{ik} = 1 \qquad \forall i = 1, \ldots, |V| \tag{5.19}$$

[8]Es wird zunächst nicht zwischen Terminvorgang und einen Vorgang mit allgemeiner Terminvorgabe unterschieden (siehe Begriffserläuterung).

Aus der allgemeinen Forderung, dass jeder Vorgang genau einer der m Touren zugeordnet werden muss, wird nun das Zeitfenster mit berücksichtigt. Im Folgenden sei die Tour T_k die Tagestour des k.ten Plantages. Ein Vorgang v_i kann nur solch einer Tagestour k zugeordnet werden, die innerhalb des Zeitfensters $[s_i, e_i]$ liegt. Ist $s_i < 0$, so kann der Vorgang ab der ersten Tour zugeordnet werden, ansonsten frühstens ab $k = s_i$. Analog muss ein Vorgang v_i bis spätestens $k = e_i$ zugeordnet werden. Gilt jedoch $e_i > m$, soll der Vorgang einer Tour $k \leq m$ zugeordnet werden.

Nun besteht jedoch die Gefahr, dass das Ausgangsproblem mit dem Modell (TT_1) und der Nebenbedingung (5.19) keine zulässige Lösung besitzt. Ursache dafür könnte sein, dass die Anzahl der Vorgänge, welche zum gleichen Zeitpunkt enden, zu groß ist und der Monteur nicht alle bis zum Ende des Zeitfensters abarbeiten kann, d. h. das Vorgangsvolumen ist größer als das Arbeitsvolumen. Tritt dieser Fall ein, muss der Dispatcher einspringen und eine Auswahl von Vorgängen entweder fremd vergeben oder manuell zu einem späteren Zeitpunkt einplanen. Das Lösungsverfahren sollte nicht abbrechen, wenn die Bedingung (5.19) nicht erfüllt werden kann.

Eine weitere Besonderheit spielen auch hier die Hauptvorgänge mit Nebenvorgängen und zwar genau die, welche ein Relativzeitintervall mit negativen Werten enthalten, also Nebenvorgänge, welche einige Tage vor dem Hauptvorgang abgearbeitet werden müssen. Gegeben sei ein Hauptvorgang v_1 mit dem Ausführungszeitintervall $[-10, 20]$ und der Nebenvorgang v_2 mit dem Relativzeitintervall $[-4, -2]$. Bei der Planung der Touren könnte der Hauptvorgang nach dem Ausführungszeitraum zur Tour T_1 hinzugefügt werden. Jedoch wäre eine Einplanung des Nebenvorgangs dann nicht mehr möglich. Somit kann der Hauptvorgang v_1 frühstens in drei Tagen eingeplant werden, d. h. es muss gelten $k > 2$. Aus diesem Grund müssen die Startzeiten s_i für jeden Hauptvorgang mit einem Nebenvorgang, welcher einige Tage vor dem Hauptvorgang ausgeführt werden, angepasst werden. Für solche Hauptvorgänge v_i mit dem Nebenvorgang v_h wird $s_i = -t_h^o + 1$ gesetzt. Dies entspricht in diesem Fall $s_1 = 3$ und somit dem Ausführungszeitintervall $[3, 20]$ für den Hauptvorgang v_1.

Wie bereits erwähnt, ist das Tourenproblem ein stark untersuchtes Problem der diskreten Optimierung. Die Literatur zu den verschiedensten Problemen ist sehr vielfältig. Jedoch sind die in der Praxis auftretenden Spezialfälle weitaus vielfältiger. So ist auch das hier beschrieben Problem der Tagestouren ein Spezialfall. In der Literatur wurden keine Modelle oder Lösungsansätze für Tourenprobleme mit Ausführungszeiträumen in Tagen gefunden. Aus diesem Grund wurden eigene Ansätze zur Modellierung und Lösung des Problems entwickelt. Im nächsten Abschnitt wird ein modifizierter Savings-Algorithmus vorgestellt, welcher eine Lösung für das Tagestourenproblem mit Beachtung der Ausführungszeiträume ermittelt.

Ermittlung einer zulässigen Lösung mit einem modifizierten Savings-Verfahren

Zunächst wird noch eine Beschränkung der Anzahl der Touren vorgenommen. Die Festlegung der Tagestouren über die gesamte Vorgangsmenge $|V'|$ ist nicht sinnvoll, da somit eine Planung der Touren über Monate vorgenommen werden müsste. Ein großer Rechenaufwand wäre erforderlich, obwohl das vollständige Ergebnis nicht benötigt wird. Durch tägliche Änderungen und Hinzukommen weiterer Vorgänge erscheint nur eine Planung von 14 Tagen [9] sinnvoll. Damit wird die Anzahl der Touren m beschränkt. Es wird im Folgenden angenommen, dass die Summe der Plandauern der Vorgangsmenge so groß ist, dass sie nicht in den nächsten m Tagen abgearbeitet werden kann.

[9] Die Beschränkung auf 14 Tage ist nur ein grobe Annahme. Eine präzise Wahl von m kann durch Tests erfolgen.

Gegeben seien die zu planenden Tage m und damit auch die Anzahl der Touren. Bei einem beschränkten m ist es nicht notwendig bzw. auch nicht möglich alle Vorgänge der Menge V einzuplanen. Es wird nur ein Teil der Menge gewählt und ein Tourenplanung mit diesem durchgeführt. Die Wahl der Teilmenge geschieht in erster Linie nach dem Ausführungszeitraum.

Beim Savings-Algorithmus (siehe Abschnitt 4.4.6) werden zunächst Pendeltouren zu jedem Vorgang erstellt und im Anschluss mit Hilfe der maximalen Savings, welche über die Gleichung (4.33) berechnet werden, Touren zusammengeschlossen. Dies wird solange durchgeführt, bis kein zulässiges Saving mehr existiert. Eine Aussage über die Reihenfolge der Touren wird nicht gegeben, da keine Bedingungen im Problem DVRP diese erfordern. Bei den Tagestouren sind die Vorgänge jedoch untereinander zum Teil abhängig und durch die Ausführungszeiträume ist die Zuordnung der Touren zu den Tagen nicht beliebig. Die Grundidee, dass durch die Savings die Zuordnungen der Vorgänge zu den Touren geschieht, bleibt erhalten nur werden die einzelnen Schritte des Savings-Verfahren geändert.

Den Ausgangspunkt bilden $m + 1$ leere Touren T_k mit $k = 1, \ldots, m + 1$. Die ersten m Touren entsprechen m aufeinander folgenden Arbeitstagen und die Arbeitszeit A ist einzuhalten. Die Menge T_{m+1} ist eine fiktive Tour und enthält am Ende alle die Vorgänge, welche den Touren T_1 bis T_m nicht zugeordnet werden konnten und somit in den m Plantagen nicht abgearbeitet werden. Die zugehörigen Aufträge zu diesen Vorgängen werden der Auftragsmenge wieder zurückgegeben und bilden die Grundlage für eine erneute Auftragszuordnung in einem späteren Planungsintervall.

Die Vorgangsmenge V enthält alle Hauptvorgänge, die in den nächsten m Tagen eingeordnet werden können, also für die gilt $s_i \leq m$ und $e_i > 0$. Zusätzlich wird eine zweite Menge NV erzeugt, welche alle zugehörigen Nebenvorgänge zur Menge V beinhaltet. Eine Indexmenge $I = \{i_{|V|+1}, \ldots, i_{|V|+|NV|}\}$ speichert die Zugehörigkeit eines Elementes aus NV zu V. Es gilt $i_h = l$, falls der h.te Nebenvorgang zum l.ten Hauptvorgang der Menge V gehört.

Die Einplanung der Nebenvorgänge erfolgt erst im Nachhinein, d. h. zunächst werden Touren für alle Hauptvorgänge der nächsten m Tage gebildet und im Anschluss die Nebenvorgänge betrachtet. Damit ein Einplanung dieser möglich ist, wird die Arbeitszeitgrenze für die Touren 1 bis m nach der Gleichung (5.18) herabgesetzt.

Im ersten Schritt werden alle Terminvorgänge der nächsten m Tage den jeweiligen Touren zugeordnet, da die Terminvorgänge oberste Priorität haben. Dabei gilt:

$$T_j = T_j \cup \{v_i\}, \text{ falls } s_i = e_i = j \qquad \forall j = 1, \ldots, m \tag{5.20}$$

Eine zeitliche Überschneidung der Terminvorgängen tritt nicht auf, da diese bereits durch die Auftragszuordnung verhindert wird (siehe Abschnitt 5.4.1).

Danach werden alle Vorgänge aus V nach ihrem Ende des Ausführungszeitraumes e_i aufsteigend sortiert und der Menge T_{m+1} zugeordnet. Jedes Element der Menge T_{m+1} spiegelt eine Pendeltour wider.

Den nach der Zuordnung (5.20) noch leeren Touren T_j für $j = 1, \ldots, m$ wird das erste Element v_i aus T_{m+1} zugeordnet, für das gilt $e_i \leq j$. Somit erhält jede Tour mindestens einen Referenzvorgang. Anstelle der Pendeltouren beim Savingsverfahren aus Abschnitt 4.4.6 bilden die Touren T_1 bis T_m nun den Ausgangspunkt für das modifizierte Savings-Verfahren. Jedoch dürfen diese m Touren *nicht* miteinander verschmolzen werden.

Im nächsten Schritt werden alle Vorgänge, welche innerhalb der nächsten m Tage enden den Touren, wenn möglich, zugeordnet. Dazu wird für jeden Vorgang $v_i \in T_{m+1}$ mit $e_i \leq m$ ein Paar von Savings zu allen Touren T_1 bis T_m berechnet. Die Berechnung der Savings erfolgt nun ähnlich zu (4.33), da jeder Vorgang $v_i \in T_{m+1}$ eine Pendeltour widerspiegelt. Beim modifizierten Savings-Verfahren werden jedoch Paare von Savings $(s_{ki}^A,\, s_{ki}^E)$ ermittelt, um in jeder Tour T_1 bis T_m deutlich zwischen ersten Vorgang und letzten Vorgang zu unterscheiden. Die Berechnung der Saving-Paare für einen Vorgang $v_i \in T_{m+1}$ und eine Tour $T_k = \{v_{k_1}, \ldots, v_{k_{|T_k|}}\}$ mit $k = 1, \ldots, m$ erfolgt in der Form:

$$(s_{ki}^A,\, s_{ki}^E) = \begin{cases} (c_{0k_1} + c_{0i} - c_{k_1 i},\, c_{k_{|T_k|}0} + c_{0i} - c_{ik_{|T_k|}}) & \text{, falls } k \in [s_i, e_i] \\ (-M_\infty, -M_\infty) & \text{, sonst.} \end{cases} \tag{5.21}$$

Enthält eine Tour bisher nur einen Vorgang, so gilt $s_{ki}^A = s_{ki}^E$. Liegt eine Tour T_k nicht im Ausführungszeitraum des Vorganges v_i ($k \notin [s_i, e_i]$), so werden die zugehörigen Savings $s_{ik}^{A \setminus E} = -M_\infty$ mit M_∞ hinreichend groß gesetzt. Somit wird kein Vorgang einer Tour T_k außerhalb des Ausführungszeitraumes zugeordnet.

Die Menge aller dieser Paare mit $e_i \leq m$ wird zu S_1 zugeordnet. Somit enthält S_1 alle Savings zu den Vorgängen, welche in den nächsten m Tagen auslaufen und somit unbedingt zuzuordnen sind. Aus der Menge S_1 wird nun das maximale Saving ermittelt. Existieren mehrere Vorgänge mit dem selben maximalen Saving, so wird der Vorgang gewählt, dessen Ausführungszeitraum eher endet.

Sei s_{lj}^A dieses maximale Saving. Bevor der Vorgang der Tour T_l hinzugefügt werden kann, muss das Saving auf Zulässigkeit geprüft werden. Ein Saving s_{lj}^A ist zulässig, wenn die Summe der Arbeitszeiten der Tour $\{v_j\} \cup T_l$ die Arbeitszeitgrenze A' nicht überschreiten. Bei Unzulässigkeit wird $s_{lj}^A = -M_\infty$ gesetzt. Ist jedoch das maximale Saving zulässig, so wird der Vorgang v_j zur Tour T_l hinzugefügt: $T_l := \{v_j\} \cup T_l$. Analog erfolgt das Vorgehen, wenn s_{lj}^E das maximale Saving ist. In diesem Fall wird der Vorgang an die Tour angehängt: $T_l := T_l \cup \{v_j\}$. Zudem werden alle zugehörigen Savings s_{hj}^A bzw. s_{hj}^E $\forall h = 1, \ldots, m$ aus der Menge S_1 entfernt.

Im Folgenden wird wieder das maximale Saving ermittelt und auf Zulässigkeit geprüft. Dies wird solange durchgeführt bis entweder die Menge $S_1 = \emptyset$ ist oder für das maximale Saving $s_{kl}^{A \setminus E} = -M_\infty \in S_1$ gilt. Ist die Menge S_1 leer, so wurden alle zeitkritischen Vorgänge, also alle Vorgänge, welche in den nächsten m Tagen auslaufen, zugeordnet. Gilt jedoch für ein maximales Saving $s_{kl}^{A \setminus E} = -M_\infty$, so existiert kein zulässiges Saving mehr. Alle zeitkritischen Vorgänge, welcher bisher nicht zugeordnet werden konnten, werden der Menge N zugewiesen. Diese Vorgänge der Menge N werden dem Dispatcher übergeben und müssen eventuell manuell eingeordnet werden oder einen neuen Ausführungzeitraum erhalten.

Im Allgemeinen werden nun jedoch die Arbeitszeiten aller Touren nicht vollständig ausgereizt sein. Aus diesem Grund werden im Anschluss alle Paare von Savings zwischen den Touren T_k mit $k = 1, \ldots, m$ und den noch in T_{m+1} vorhandenen Vorgängen, deren Abarbeitung in den m Plantagen nicht zwingend ist, nach (5.21) berechnet. Diese Paare werden der Menge S_2 zugeordnet. Das maximale Saving $s_{lj}^{A \setminus E}$ wird bestimmt. Ist dieses zulässig, dann wird der Vorgang v_j der Tour T_l hinzugefügt. Überschreitet diese neue Tour die Arbeitszeitgrenze, d. h. sie ist nicht zulässig, dann wird auch hier das maximale Saving auf $-M_\infty$ gesetzt.

Der Algorithmus wird solange durchgeführt bis für das maximale Saving $s_{lj}^{A \setminus E} = -M_\infty$ gilt und damit kein zulässiges Saving mehr existiert. Im Anschluss werden noch die zugehörigen Nebenvorgänge zu den Hauptvorgängen der nächsten m Tage zugeordnet. Dazu werden für jeden Nebenvorgang die Savings zu allen Touren in der Relativzeit nach (5.21) berechnet. Der Nebenvorgang wird der Tour mit dem maximalen Saving zugeordnet. Auf Zulässigkeit wird in diesem Fall nicht mehr geprüft.

Im Folgenden wird der Algorithmus an einem Beispiel nachvollzogen. Es seien die nächsten 3 Tage zu planen $(m = 3)$, die Arbeitszeitgrenze, welche sich nach (5.18) mit $A = 9\,h$ ergibt, beträgt $A' = 8h\,59min$ und die folgende Tabelle gibt die vorhandenen Vorgänge mit ihren wichtigen Informationen an.

Vorgang	Nebenvorgang mit Relativzeit und Plandauer	Ausführungs-zeitraum	Plandauer	Fahrzeit zum Depot
v_1		[1, 1]	$3\,h$	$45\,min$
v_2		[2, 2]	$5\,h$	$30\,min$
v_3		[-10, 1]	$6\,h30\,min$	$45\,min$
v_4		[-50, 2]	$1\,h\;30\,min$	$25\,min$
v_5		[- 2, 3]	$3\,h$	$45\,min$
v_6	v_{16} mit [1,2] und $30\,min$	[2,10]	$2\,h$	$15\,min$
v_7		[-30,10]	$6\,h$	$1\,h$
v_8		[-30,10]	$5\,h$	$10\,min$
v_9		[-30,10]	$45\,min$	$35\,min$
v_{10}		[-20,20]	$1\,min$	$1\,h$
v_{11}		[-15,40]	$30\,min$	$35\,min$
v_{12}		[-10,50]	$3\,h$	$50\,min$
v_{13}		[-4,55]	$1\,h\;30\,min$	$50\,min$
v_{14}		[-2,55]	$2\,h\;30\,min$	$45\,min$
v_{15}		[0,90]	$1\,h\;30\,min$	$20\,min$

Auf konkrete Werte der Fahrzeiten zwischen den Vorgängen wird verzichtet. Die Lage der Orte und damit die Relationen der Fahrzeiten zwischen den einzelnen Vorgängen können aus der Abbildung 5.5a) entnommen werden.

Im ersten Schritt werden die Terminvorgänge v_1 und v_2 zu ihren Tour hinzugefügt. Da nur 2 Terminvorgänge vorhanden sind, wird der dritten Tour, in diesem Beispiel T_3, der Vorgang mit dem kleinsten $e_i > 3$ zugeordnet. Dies entspricht dem Vorgang v_5. Das Resultat mit $T_1 = \{v_1\}$, $T_2 = \{v_2\}$ und $T_3 = \{v_5\}$ ist in Abbildung 5.5a) zu sehen. Die Vorgänge v_1, v_2, und v_5 werden aus der Tabelle gestrichen bzw. werden im Folgenden nicht mehr beachtet.

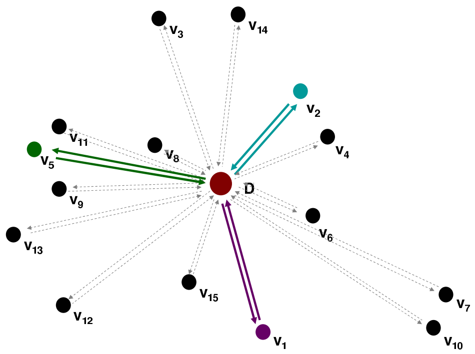
Im nächsten Schritt werden die Vorgänge herausgefiltert, welche in den nächsten 3 Tagen ablaufen. Aus der Tabelle ist zu entnehmen, dass dies für die Vorgängen v_3 und v_4 mit $e_i \leq 3$ gilt. Für diese zwei Vorgänge werden nun die Savings-Paare zu den Touren T_1 bis T_3 berechnet, wobei für alle gilt $s^A_{ij} = s^E_{ij}$, da bisher nur ein Vorgang pro Tour enthalten ist. Zur Berechnung der Savings werden die Fahrzeiten zwischen den Vorgängen v_3, v_4 und v_1, v_2, v_5 benötigt. Diese und die zugehörigen Savings sind in den zwei folgenden Tabellen aufgelistet:

Fahrzeiten zwischen den Vorgängen:

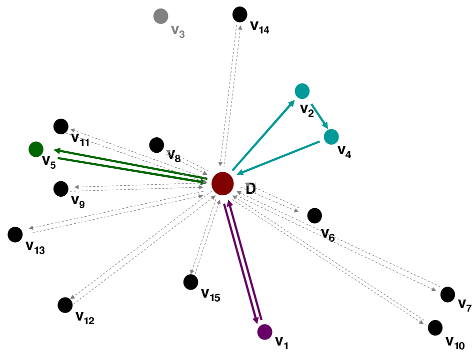
	v_3	v_4
v_1	1 h 90 min	1 h
v_2	45 min	10 min
v_5	1 h	1 h 5 min

Zugehörige Savings:

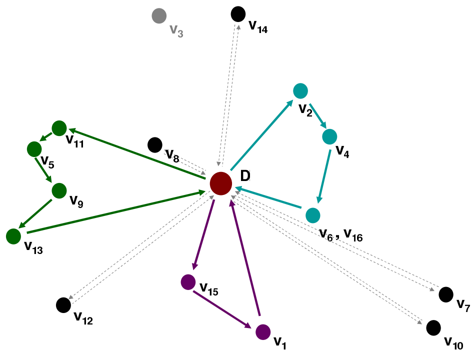
	v_3	v_4
$T_1 = \{v_1\}$	0 min	10 min
$T_2 = \{v_2\}$	M_∞	1 h
$T_3 = \{v_5\}$	M_∞	M_∞



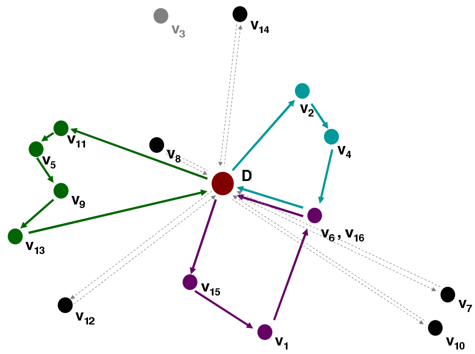
a) Zuweisung der Terminvorgänge zu den Touren



b) Zuweisung der zeitkritischen Vorgänge



c) Ergebnis für die Hauptvorgänge



d) Endergebnis nach Zuweisung der Nebenvorgänge

Abbildung 5.5: Beispiel für eine Tourenbildung mit Hilfe des modifizierten Savings-Verfahren

Die Savings, berechnet nach (5.21), werden der Menge S_1 zugewiesen. Als erstes maximales Saving ergibt sich $s_{24}^{A\setminus E} = 1\,h$. Dieses Saving ist zudem zulässig, da die Arbeitszeitkapazität nicht überschritten wird. Der Vorgang v_4 wird der Tour $T_2 = \{v_2\}$ angeschlossen. Alle Savings, welche zum Vorgang v_4 gehören, werden entfernt. Die damit sich ergebende Änderungen in der Tour $T_2 = \{v_2, v_4\}$ ist in der Abbildung 5.5b) dargestellt.

Das neue maximale Saving ist s_{31}. Dieses Saving ist jedoch nicht zulässig, da die Arbeitszeitgrenze A' bereit durch die Summe der Plandauern $p_3+p_5 = 9\,h\;30\,min$ überschritten wird. Somit wird s_{31} auf M_∞ gesetzt. Da nun das maximale Saving den Wert M_∞ besitzt, existiert keine zulässige Tour für den Vorgang v_3. Damit wird der Vorgang v_3 der Menge N hinzugefügt und an den Dispatcher zur manuellen Einplanung übergeben.

Alle zeitkritischen Vorgänge wurden somit bearbeitet. Im Anschluss werden alle weiteren Vorgänge betrachtet und eingeordnet. Das Ergebnis der Tourenplanung für die Hauptvorgänge ist in der Abbildung 5.5c) zusehen.

Am Ende werden noch alle Nebenvorgänge bearbeitet. Der Hauptvorgang v_6 besitzt den Nebenvorgang v_{16}. Da v_6 der zweiten Tour zugeordnet wurde ist es für den Nebenvorgang nur noch möglich in die erste Tour T_1 hinzugenommen zu werden. Somit werden den Touren die folgenden Vorgänge zugeordnet: $T_1 = \{v_{15}, v_1, v_{16}\}$, $T_2 = \{v_2, v_4, v_6\}$ und $T_3 = \{v_{11}, v_5, v_9, v_{13}\}$. Das Endergebnis ist ebenso aus der Abbildung 5.5d) zu entnehmen.

Die Menge der nicht automatisch zu zuordnenden Vorgänge ist $N = \{v_3\}$. Diese wird dem Dispatcher zu manuellen Einplanung übergeben. Die Menge $T_4 = \{v_7, v_8, v_{10}, v_{12}, v_{14}\}$ enthält nach Bildung der Touren alle die Vorgänge, welche in den nächsten 3 Tagen nicht abgearbeitet werden. Die zugehörigen Aufträge zu den Vorgängen der Menge T_4 werden wieder der Auftragszuordnung übergeben und bei der nächsten Planung mit berücksichtigt.

Verbesserungsverfahren

Der modifizierte Savings-Algorithmus ermittelt eine zulässige Lösung für die Tagestouren. Diese liefert eine gute Grundlage für Verbesserungsverfahren, wie zum Beispiel das 2 bzw. 3-opt-Verfahren.

Das 2 bzw. 3-opt-Verfahren kann analog wie im Abschnitt 4.4.7 angewandt werden. Nur der Begriff der Zulässigkeit ist weiter gespannt. Eine Vertauschung der Kanten ist nur dann zulässig, wenn neben der maximalen Arbeitszeit auch die Ausführungszeiträume beachtet werden. Eine Verbesserung ist nur dann gegeben, wenn diese innerhalb der ersten m Touren stattfindet. Eine Verbesserung innerhalb der Tour T_{m+1} ist nicht sinnvoll, da diese keine reale Tour ist. Somit braucht eine toureninterne Verbesserung nur innerhalb der Touren T_1 bis T_m durchgeführt werden. Bei der tourenübergreifenden Verbesserung kann die Tour T_{m+1} mit einbezogen werden.

Tourenintern kann auch ein exaktes Verfahren wie zum Beispiel das Branch&Bound-Verfahren für TSP angewendet werden. Ein exaktes Verfahren ist hier möglich, da die Anzahl der Vorgänge pro Tour im Allgemeinen nur gering ist.

5.3.4 Erweiterung: Uhrzeitfixe Terminvorgänge

Bei envia NSG wird in Vorgängen mit Priorität „allgemeine Terminvorgabe“ und „Terminvorgang“ unterschieden (siehe Kapitel 2). Die Besonderheit bei den Terminvorgängen ist, dass diese nicht nur einen konkreten Tag zugeordnet werden, wie die Vorgänge mit „allgemeiner Terminvorgabe“, sondern ebenso eine Uhrzeit für den Beginn des Vorganges enthalten. Somit können Vorgänge nicht beliebig davor oder danach geschalten werden. Ist innerhalb einer Tour ein Terminvorgang enthalten, muss bei der Tour mit dem maximalen Saving beachtet werden, ob der neue Vorgang überhaupt vorher oder nachher eingeordnet werden kann.

Somit muss bei der Tourenplanung, neben der Bearbeitungsreihenfolge, ebenso die Startzeit jedes einzelnen Vorgangs ermittelt werden. Dazu ist es notwendig ein Tagesarbeitszeitraum, z. B. [6:00, 18:00][10] festzulegen. So kann ein Monteur frühstens ab 6:00 Uhr seine Arbeit aufnehmen und darf bis maximal 18:00 Uhr arbeiten. Jedoch darf dieser an einem Tag nicht von 6:00 Uhr bis 18:00 ausgelastet sein, da dies der maximalen Arbeitszeit, hier von 9 Stunden, widerspricht. Die Beachtung der Uhrzeiten bei Terminvorgängen macht den Algorithmus weit aus komplexer. Der Begriff der Zulässigkeit für ein Saving muss noch weiter gefasst werden.

Sind einem Tag bzw. einer Tour keine Terminvorgänge zugeordnet, so spielen Uhrzeiten keine Rolle. Bei diesen Tour muss jeglich auf die Einhaltung der Arbeitszeitgrenze geachtet werden. Nur bei Touren mit Terminvorgängen bildet der Tagesarbeitszeitraum eine neue Bedingung. Für jede Tour mit Terminvorgang wird der Zeitpunkt für den Start der Vorgänge und für das Ende der Vorgänge ermittelt. Es kann nur dann ein Vorgang in einer Tour noch davor geschaltet werden,

[10] Auch diese Angaben sind eine Annahme und können jeder Zeit neu gesetzt werden.

wenn die so neu entstehende Tour nicht vor 6:00 Uhr beginnt. Analog kann an die Tour nur dann ein Vorgang angeschlossen werden, wenn die so neu entstehende Tour vor 18:00 Uhr endet.

Dazu ein kleines Beispiel: Die Tour T_k besitzt einen Terminvorgang v_1, welcher 8:00 Uhr beginnt und eine Plandauer von 4 Stunden besitzt. Zur Planung stehen fünf weitere Vorgänge $v_2, \ldots, v_6$ zur Verfügung, welche keine Terminvorgänge sind:

Vorgang (v_i)	Plandauer (p_i)	Fahrzeit zum Vorgang v_1 (f_{i1})	Fahrzeit zum Depot (f_{i0})
$\boldsymbol{v_1}$	**240 min**	**0 min**	**25 min**
v_2	120 min	25 min	20 min
v_3	200 min	15 min	30 min
v_4	360 min	15 min	15 min
v_5	80 min	10 min	15 min
v_6	20 min	10 min	15 min

Die Vorgänge v_5 und v_6 seien zwei Inspektionen in einem Ortsteil und die Fahrzeit zwischen diesen beiden Inspektionen ist gleich Null. Die Lage der Vorgänge zueinander und damit die Fahrzeiten zwischen diesen können aus der Abbildung 5.8a) abgeschätzt werden.

Der Terminvorgang v_1 liegt fest von 8:00 Uhr bis voraussichtlich 12:00 Uhr. Nun steht die Frage, welcher der Vorgänge v_2 bis v_6 vorher und welche nachher eingeplant werden kann. Die Abbildung 5.6 verdeutlicht, welche Einplanungen möglich sind und welche nicht, da sie die Arbeitszeitgrenzen überschreiten würden. Die Vorgänge v_5 und v_6 besitzen eine relativ kleine Plandauer. Es ist zulässige diese sowohl vorher, als auch nachher einzuplanen. Bei den Vorgängen v_2 und v_3 ist eine Zuordnung vor dem Terminvorgang nicht möglich, da der Monteur dann vor 6:00 Uhr vom Depot starten müsste. Jedoch können diese im Anschluss an v_1 durchgeführt werden, da der Monteur dann vor 18:00 Uhr zum Depot zurückkehren würde und auch die maximale Arbeitszeitgrenze nicht überschritten wird. Der Vorgang v_4 kann in die Tour T_k nicht eingefügt werden, da dieser sowohl vorher als auch nachher die Arbeitszeitgrenzen überschreiten würde.

Es sei nun der Vorgang v_5 bereits zur Tour hinzugenommen, so dass $T_k = \{v_5, v_1\}$ gilt. Damit ist es nicht möglich außer dem Vorgang v_5 auch noch v_6 dem Vorgang v_1 vorzuschalten, da der Monteur in diesem Fall 5:50 Uhr starten müsste. Würden an das Ende der Tour die Vorgänge v_2, v_3 oder v_6 angeschlossen werden, dann wäre jeweils die Tour vor 18:00 Uhr beendet. Da die Summe der Arbeitszeiten der Vorgänge v_1, v_5 und v_3 die Arbeitszeitgrenze von 9 Stunden jedoch überschreiten würde, sind nur die Zuordnung $T_k = \{v_5, v_1, v_2\}$ oder $T_k = \{v_5, v_1, v_6\}$ in diesem Beispiel möglich. Verdeutlicht wird dies noch einmal in der Abbildung 5.7.

Der Vorgang v_6 bringt bei Hinzunahme die größte Ersparnis und damit wird dieser Vorgang der Tour angeschlossen. Die Tour $T_k = \{v_5, v_1, v_6\}$ besitzt nun eine Arbeitszeit von $6h\,30min$. Der Vorgang v_2, kann der Tour T_k nicht angeschlossen werden, da die Fahrzeit zwischen v_2 und v_6 $45\,min$ beträgt und damit die Gesamtarbeitszeit $9h\,25min$, was die Arbeitszeitgrenze überschreiten würde. Damit können weitere Vorgänge zur Tour T_k nicht hinzugefügt werden. Eine längere Arbeitszeit, jedoch auch eine höhere Fahrzeit würde die Tour $T_k = \{v_5, v_1, v_2\}$ erzielen. Hier beträgt die Arbeitszeit $8h\,10min$, diese Tour könnte später durch die Anwendung des Verbesserungsverfahren bevorzugt werden. Dies genannten Touren $T_k = \{v_5, v_1, v_6\}$ und $T_k = \{v_5, v_1, v_2\}$ werden grafisch in der Abbildung 5.8b) und 5.8c) verdeutlicht.

Bereits an dem kleinen Beispiel wird der Umfang und die Einschränkung der Zuordnungen deutlich, welcher zusätzlich durch die uhrzeitfixe Festlegung der Termine entsteht. Wären zunächst keine Uhrzeiten für die Terminvorgänge wie bei den Vorgängen mit „allgemeiner Terminvorgabe“

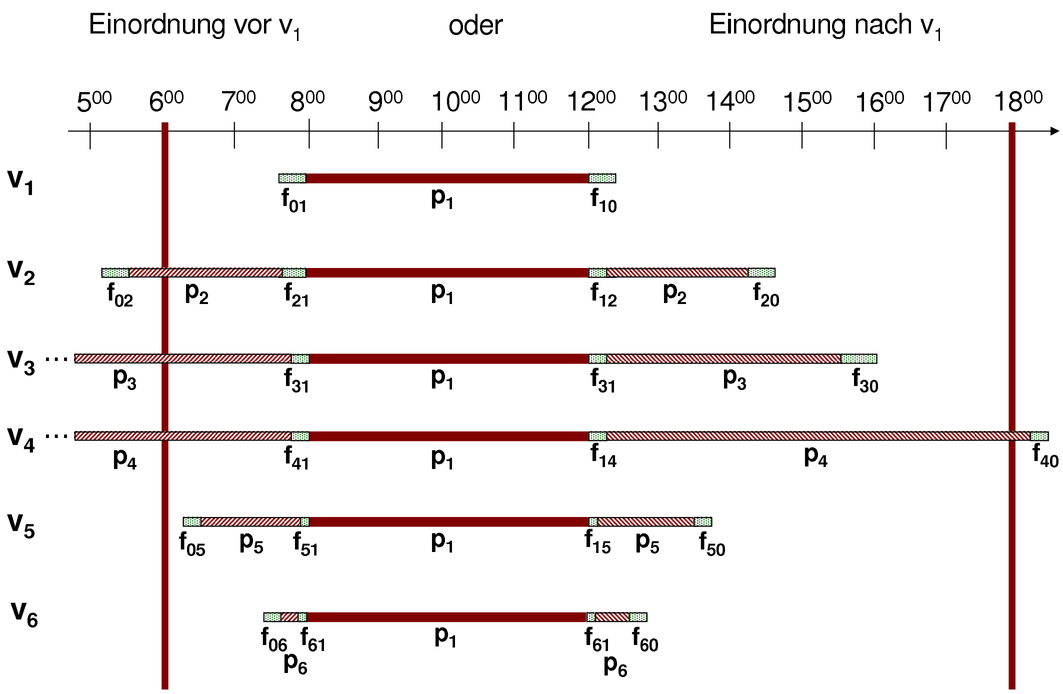


Abbildung 5.6: Zeitstrahl bei Einplanung der Vorgänge vor oder nach Vorgang v_1

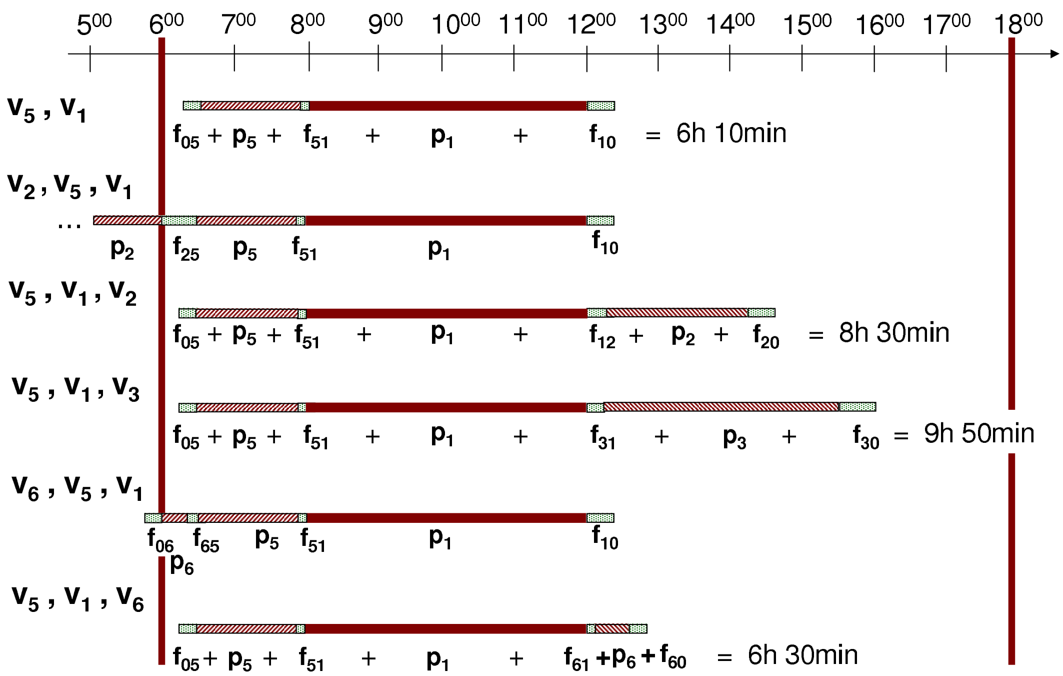
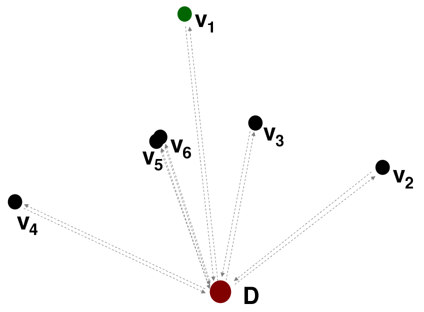
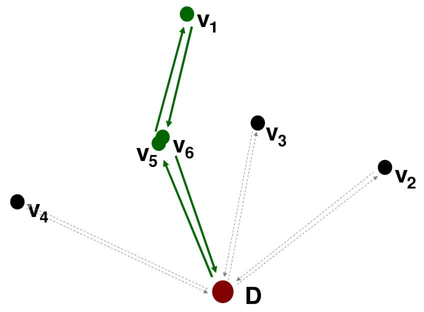


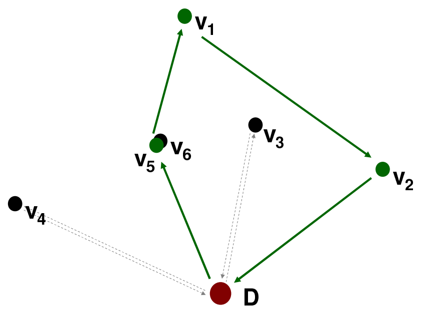
Abbildung 5.7: Zeitstrahl bei Einplanung der Vorgänge zur Tour $T_1 = \{v_5, v_1\}$



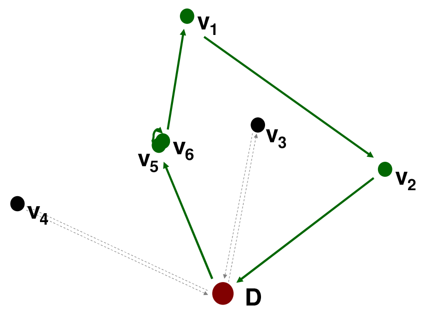
a) Ausgangsproblem mit v_1 als Terminvorgang



b) Zulässige Tour $T_k = \{v_5, v_1, v_6\}$ mit modifizierten Savings-Algorithmus



c) Weitere zulässige Tour $T_k = \{v_5, v_1, v_2\}$



d) Tour $T_k = \{v_6, v_5, v_1, v_2\}$ **ohne** feste Uhrzeit der Terminvorgänge

Abbildung 5.8: Terminvorgänge im modifizierten Savings-Verfahren

gegeben, so ergibt sich bei diesem Beispiel die Tour $T_k = \{v_6, v_5, v_1, v_2\}$ mit der Arbeitszeit von $8\,h\;\;30\,min$, wie in Abbildung 5.8d) dargestellt ist. Damit könnte in der Tour ein Vorgang mehr ausgeführt werden. Die Startzeit für den Terminvorgang könnte nach dem Durchlauf der Optimierung auf zum Beispiel 9:00 Uhr festgesetzt werden oder, da die Tagestour auch in umgekehrter Reihenfolge abgearbeitet werden kann auf 11:00 Uhr.

Die uhrzeitfixe Restriktion muss sowohl als Bedingung im Modell als auch beim Lösungsverfahren berücksichtigt werden. Zur Modellierung wird zusätzlich ein Parameter benötigt, welcher die Uhrzeiten festhält. Jeder Vorgang v_i erhält u_i als Startzeitpunkt. Sind in einer Tour keine Terminvorgänge zugeordnet, so wird am Ende u_j =7:00 Uhr gesetzt, wobei v_j der erste Vorgang dieser Tour ist. Alle weiteren Vorgänge werden mit Beachtung der Plandauer und Fahrzeit nahtlos angereiht. Somit ist der späteste Zeitpunkt zur Rückkehr unter Einhaltung der Arbeitszeitgrenze vom letzten Vorgang der Tagestour zum Depot 16:00 Uhr. Diese Festlegung dient dazu, dass u_i für alle Vorgänge definiert ist. Ist in der Tour ein Terminvorgang enthalten, werden ausgehend von diesem die Uhrzeiten der weiteren Vorgänge berechnet.

Im linearen Modell TT_1 müssen folgende zwei Nebenbedingung zusätzlich aufgenommen werden, welche gewährleisten, dass der tägliche Tagesarbeitszeitraum eingehalten wird:

$$u_i \geq 6:00 \qquad \forall i : y_{ik} > 0 \tag{5.22}$$

$$u_i + \frac{1}{2}p_i + c_{i0} \leq 18:00 \qquad \forall i : y_{ik} > 0 \tag{5.23}$$

In der Berechnung des voraussichtlichen Endes eines Vorganges muss neben der Fahrzeit c_{i0} nur die Hälfte der Plandauer des Vorganges dazu addiert werden, da eine Hälfte bereits in der Arbeitszeit c_{i0} enthalten ist (siehe (5.13)). Das Hinzufügen der beiden Nebenbedingungen aus (5.23) vergrößert das Modell (TT_1) nochmals und führt damit zu einem größeren Speicherplatzbedarf.

Im modifizierten Savings-Algorithmus wird dabei eine weitere Bedingung zur Überprüfung der Zulässigkeit hinzugenommen. Es muss beim maximalen Saving geprüft werden, ob nach dem Hinzufügen die Uhrzeiten der Vorgänge der Tour innerhalb des Tagesarbeitszeitraum liegen. Dies führt zu einer aufwändigen Speicherung und damit zu einer Erhöhung der Rechenzeit.

Bisher wurden nur Touren betrachtet, welche einen Terminvorgang enthalten. Es ist zudem möglich, dass mehrere Terminvorgänge in einer Tour enthalten sind. Eines der betrieblichen Ziele ist das Füllen der Einsatzliste bei bekannten Terminvorgängen (siehe BZ 5) mit Hilfe weitere Vorgänge aus der Vorgangsmenge. Durch die festen Uhrzeiten entstehen zwischen zwei Terminvorgängen *Lücken*. Das Ziel ist es, diese mit einem möglichst passenden Vorgang zu füllen.

Definition 9 (Lücke). *Eine Lücke l_{ij} ist die zeitliche Differenz zwischen dem voraussichtlichen Ende eines Vorganges v_i und dem Planbeginn des darauf folgenden Terminvorgangs v_j einer Tour.*

Eine Lücken kann zum Beispiel dann auftreten, wenn zum Ausführen eines Vorganges Nebenvorgänge, wie eine Hin- und Rückschaltung, erforderlich sind und der eigentliche Hauptvorgang von einer Fremdfirma erledigt wird. Die Schaltungen sind Terminvorgänge, welche vom Monteur erledigt werden. Zwischen den Schaltungen hat der Monteur jedoch Freiraum.

Innerhalb des linearen Modells kann diese Forderung zur Zielfunktion mit hinzugenommen werden. Dabei wird die Zeit zwischen dem voraussichtlichen Ende des Vorganges v_i und dem Beginn des Terminvorgangs v_j berechnet:

$$l_{ij} = \begin{cases} u_j - (u_i + p_i) & \text{, falls } v_j \text{ ein Terminvorgang ist} \\ 0 & \text{, sonst.} \end{cases} \tag{5.24}$$

Die Berechnung der Lücke wird nur für toureninterne Vorgänge vorgenommen und erfolgt ohne Beachtung der Fahrzeit.

Sollen die Lücken zwischen zwei Terminvorgängen minimiert werden, ergibt sich in der Zielfunktion aus (TT_1) ein weiterer Term, sodass folgende neue Zielfunktion entsteht:

$$\text{ZF':} \qquad \sum_{k=1}^{m}\sum_{i=0}^{|V|}\sum_{j=0}^{|V|}[c_{ij} + \alpha\, l_{ij}]x_{ijk} \rightarrow min \tag{5.25}$$

Mit Hilfe des Strafparameters α kann der Einfluss der Lückenminimierung auf die Zielfunktion geregelt werden. Je größer α ist, desto stärker wirkt eine große Lücke gegen die Optimierungsrichtung.

Zudem muss beachtet werden, dass $l_{ij} > 0$ nur dann gilt, wenn zwei Terminvorgänge in einer Tour enthalten sind. Zwischen diesen beiden Terminvorgängen werden weitere Vorgänge gesucht, die diese Lücke möglichst vollständig schließen. Da ein perfekter Vorgang v_h, also ein Vorgang für den gilt: $l_{ij} = f_{ih} + p_h + f_{hj}$ mit f_{ij} als Fahrzeit zwischen den Ausführungsorten von v_i und v_j, in der Regel nicht vorhanden sein wird, bleibt im Allgemeinen ein $l_{hj} > 0$ übrig. Die Wartezeit zwischen zwei Vorgängen gehört zur Arbeitszeit und aus diesem Grund muss die Zeit bei der Berechnung der Tagesarbeitszeit mit berücksichtigt werden. Die Nebenbedingung (5.28b) muss deshalb leicht modifiziert werden:

$$\sum_{i=0}^{|V|} \sum_{j=0}^{|V|} (c_{ij} + l_{ij}) x_{ijk} \leq A' \qquad \forall k = 1, \ldots, m \tag{5.26}$$

Im Eröffnungsverfahren werden bisher Vorgänge immer nur unmittelbar an einen Vorgang davor gesetzt oder angehangen. Sollen nun die Lücken zwischen zwei Terminvorgängen möglichst gut geschlossen werden, so ist ein weiterer Schritt im Algorithmus nötig. Nachdem alle Terminvorgänge den nächsten m Tagestouren zugeordnet wurden, werden die Touren T_k herausgefiltert, welche mehr als einen Terminvorgang besitzen. Zwischen zwei Terminvorgängen wird die Lücke l_{ij} berechnet. Im Anschluss wird für jeden einzelnen Vorgang $v_h \in T_{m+1}$ die Lücke l'_{hj} wie folgt berechnet:

$$l'_{hj} = \begin{cases} l_{ij} - (f_{ih} + p_h + f_{hj}) & \text{, falls } l_{ij} > (f_{ih} + p_h + f_{hj}) \\ & \text{ und für } v_i, v_j \in T_k : s_h \leq k \leq e_h \\ M_\infty & \text{, sonst} \end{cases} \tag{5.27}$$

wobei M_∞ hinreichend groß ist und f_{ij} der Fahrzeit zwischen den Ausführungsorten der Vorgänge v_i und v_j. Der Wert l'_{hj} entspricht der neuen Lücke, welche durch Hinzufügen des Vorganges v_h zwischen die Terminvorgänge v_i und v_j entsteht. Es gilt: $l'_{hj} < l_{ij}$, wenn der Vorgang v_h zeitlich zwischen die Terminvorgänge passt. Ist die Plandauer und Fahrzeit des Vorganges v_h größer als die Lücke oder liegt die Tour T_k nicht im Ausführungszeitraum von v_h, so ist dies nicht zulässig und l'_{hj} wird auf M_∞ gesetzt. Die Fahrzeiten f_{ij} können aus den Arbeitszeiten $c_{ij} = \frac{1}{2}p_i + f_{ij} + \frac{1}{2}p_j$ ermittelt werden. In der Abbildung 5.9 werden die Fälle für eine Lücke l'_{hj} eines Vorgang v_h aufgezeigt.

Im Anschluss wird der Vorgang v_l gewählt, welcher eine minimale Lücke l'_{lj} besitzt. Im optimalen Fall ist $l'_{lj} = 0$. In diesem Fall würde die Lücke vollständig geschlossen werden. Gilt für die minimale Lücke $l'_{lj} = M_\infty$, dann steht kein Vorgang zu Verfügung, welcher zwischen beide Terminvorgänge passt.

Bei der Berechnung in (5.27) wird immer nur ein Vorgang zu Schließung der Lücke verwendet. Eine weitere Verbesserung könnte die Verwendung von mehreren Vorgängen sein. Dazu müssten alle Kombinationen von Vorgängen bestimmt werden, welche eine Gesamtplandauer mit Gesamtfahrzeit kleiner als die Lücke l_{ij} besitzen. Der Aufwand der Bestimmung dieser Kombinationen ist jedoch immens und damit steigt der Rechenaufwand. Ob die Verwendung von mehreren Vorgängen zu Schließung der Lücken einen größeren Nutzen bringt, welche den Aufwand rechtfertigt, muss in der Praxis getestet werden. Innerhalb dieser Arbeit wird zunächst nur ein Vorgang zur Lückenschließung verwendet.

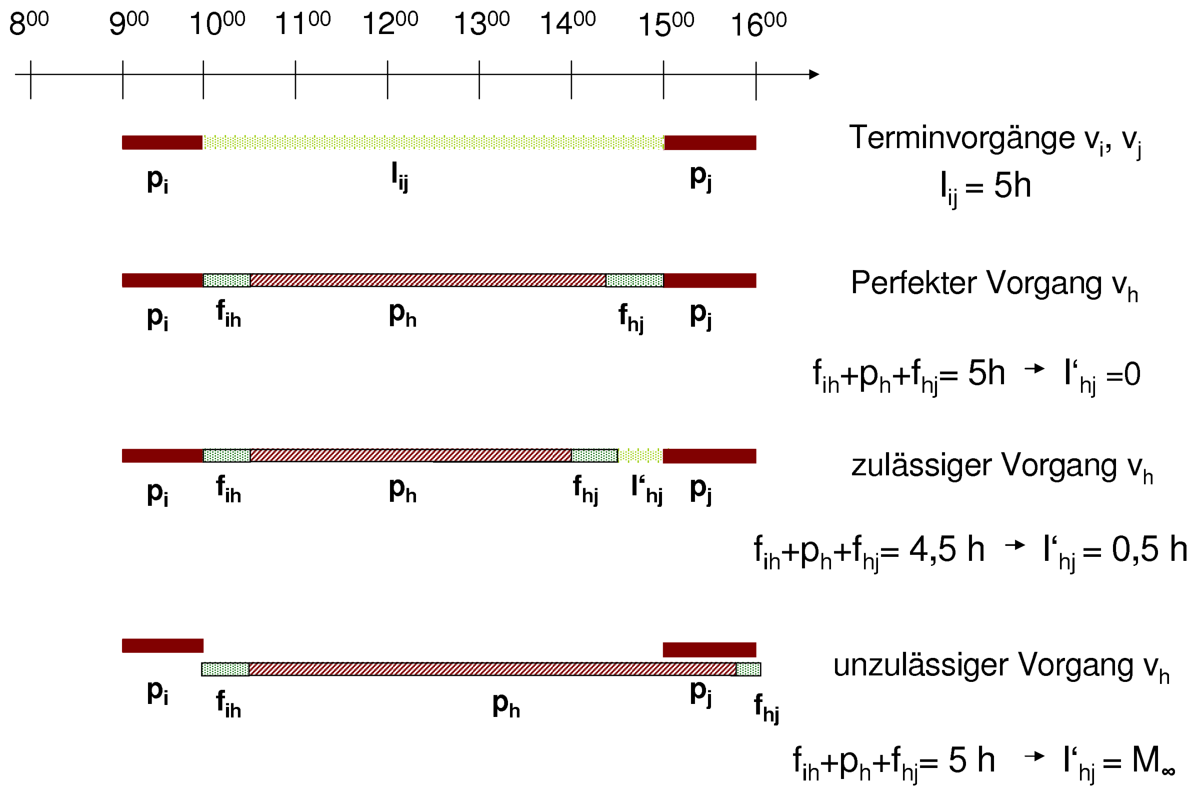


Abbildung 5.9: Drei Fälle für Vorgang v_h bei der Lückenschließung

5.3.5 Gesamtmodell

In diesem Teilkapitel wurde zunächst von einen vereinfachten Problem der Tagestouren ausgegangen. Dieses konnte auf das wohlbekannte Problem der Tourenplanung zurückgeführt werden. In den nächsten Abschnitten wurden die Besonderheiten bei den Tagestouren erläutert und Vorschläge für die Modellanpassung gegeben.

Das folgende Modell (TT_2) fasst die Erweiterungen des Modells (TT_1), welche in den Abschnitten 5.3.2 bis 5.3.4 erläutert wurden, zusammen.

(TT_2)

$$\text{ZF:} \quad \sum_{k=1}^{m}\sum_{i=0}^{|V|}\sum_{j=0}^{|V|}(c_{ij}+\alpha\, l_{ij})\, x_{ijk} \quad \rightarrow min \tag{5.28a}$$

$$\text{unter den NB:} \quad \sum_{i=0}^{|V|}\sum_{j=0}^{|V|}(c_{ij}+l_{ij})\, x_{ijk} \leq A' \quad \forall k=1,\ldots,m \tag{5.28b}$$

$$\sum_{k=1}^{m} y_{ik} = 1 \quad \forall i=1,\ldots,|V| \tag{5.28c}$$

$$\sum_{i=0}^{|V|} x_{ijk} \leq y_{ik} \quad \forall k=1,\ldots,m;\ j=0,\ldots,|V| \tag{5.28d}$$

$$\sum_{j=0}^{|V|} x_{ijk} \leq y_{jk} \quad \forall k=1,\ldots,m;\ i=0,\ldots,|V| \tag{5.28e}$$

$$\sum_{(i,j)\in S} x_{ijk} \leq |S|-1 \quad S\subset V, |S|\geq 1, k=1,\ldots,m \tag{5.28f}$$

$$y_{i_h k} - \sum_{s\in[t_h^u,t_h^o]} y_{hk+s} = 0 \quad \forall i_h \in I \tag{5.28g}$$

$$\sum_{k=\max\{s_i,m\}}^{\min\{e_i,m\}} y_{ik} = 1 \quad \forall i=1,\ldots,|V| \tag{5.28h}$$

$$u_i \geq 6:00 \quad \forall i: y_{ik}>0 \tag{5.28i}$$

$$u_i + \frac{1}{2}p_i + c_{i0} \leq 18:00 \quad \forall i: y_{ik}>0 \tag{5.28j}$$

$$x_{ijk}\in\{0,1\} \quad \forall\, i,j=0,\ldots,|V|;\ k=1,\ldots,m \tag{5.28k}$$

$$y_{0k}=1 \wedge y_{ik}\in\{0,1\} \quad \forall\, i=1,\ldots,|V|\, k=1,\ldots,m \tag{5.28l}$$

Die Forderung der optimalen Lückenschließung erfolgt durch die zusätzlichen Terme in der Zielfunktion (5.28a) und der ersten Nebenbedingung (5.28b). Zudem wurde die Nebenbedingung (5.28g) zur Einhaltung der Relativzeiten der Nebenvorgänge zu den Hauptvorgängen hinzugefügt. Die Nebenbedingung (5.28h) veranlasst, dass jeder Vorgang innerhalb des Zeitfensters eingeordnet wird. Für die Einhaltung der Tagesarbeitszeiten dienen die Nebenbedingungen (5.28i) und (5.28j).

Verbal wurde die Veränderungen im modifizierten Savings-Algorithmus bereits an entsprechender Stelle erläutert. An dieser Stelle wird dieser noch einmal zusammengefasst:

Modifizierter Savings-Algorithmus zur Tagestourenplanung:

Gegeben: Gegeben sei die Anzahl der Touren m und die Menge der Vorgänge V, welche in den m Plantagen eingeordnet werden können, sowie die zugehörigen Nebenvorgänge NV mit der Indexmenge I.

Initialisierung: Setze die Menge der Touren $T_k = \emptyset \; \forall k = 1, \ldots, m+1$ und die Menge der nicht automatisch einordenbaren Vorgänge $N = \emptyset$.

Schritt 1: Berechne A' nach (5.18).

Schritt 2: Ordne die Terminvorgänge $v_i \in V$ der m Tage zu den Touren nach (5.20) zu. Entferne diese zugeordneten Vorgänge aus V.

Schritt 3: Ermittle alle Touren, welche mehr als einen Terminvorgang pro Tag besitzen. Ordne diese der Menge W zu.

Schritt 4: Gilt $W = \emptyset$?

nein Wähle eine Tour $T_k \in W$. Berechne die Lücke zwischen den Terminvorgänge l_{ij} nach (5.24). Berechne l'_{hj} für alle $v_h \in V$ nach (5.27). Ermittle die minimale Lücke l'_{lj} mit $s_l \leq k \leq e_l$ und füge v_l in die Tour zwischen v_i und v_j. Entferne v_l aus T_{m+1}, setze $W := W \backslash T_k$ und gehe zu Schritt 4.

ja Gehe weiter zu Schritt 5.

Schritt 5: Sortiere die Vorgangsmenge V nach dem Ende der Ausführungszeiträume ihrer Vorgänge und ordne die sortiere Menge der Menge T_{m+1} zu.

Schritt 6: Existiert ein $k < m + 1$ mit $T_k = \emptyset$?

nein Gehe weiter zu Schritt 7.

ja Setze $T_k = \{v_i\}$, wobei v_i das erste Element aus T_{m+1} ist, für das gilt $e_i \leq k$. Entferne v_i aus T_{m+1}.

Schritt 7: Berechne die Menge aller Savings $S_1 = \{s_{kl}\}$ nach (5.21) zu den Touren T_k mit $k \leq e_l$ und den Vorgängen v_l mit $e_l \leq m$.

Schritt 8: Gilt $S_1 = \emptyset$?

nein Gehe zu Schritt 9.

ja Es existiert keine Vorgang v_l mit $e_l \leq m$, welcher noch keiner Tour T_1 bis T_m zugeordnet wurde. Gehe weiter zu Schritt 12.

Schritt 9: Ermittle das maximale Saving $s_{jl} \in S_1$.

Schritt 10: Gilt $s_{jl} = -M_\infty$?

nein Gehe zu Schritt 11.

ja Alle Vorgänge v_l mit $e_l \leq m$, welche bisher nicht zugeordnet wurden, können nicht automatisch eingeordnet werden. Entferne diese aus T_{m+1} und füge sie zur Menge N hinzu. Gehe zu Schritt 12.

Schritt 11: Ist das Saving s_{jl} zulässig?

nein $s_{jl} = -M_\infty$. Gehe zu Schritt 9.

ja Füge den Vorgang v_l zur Tour T_j hinzu. Setze $T_{m+1} = T_{m+1} \backslash v_j$ und entferne alle Savings die zum Vorgang v_l gehören aus S_1. Gehe zu Schritt 8.

Schritt 12: Berechne die Menge aller Savings $S_2 = \{s_{ki} \,|\, k = 1, \ldots, m; v_i \in T_{m+1}\}$.

Schritt 13: Ermittle das maximale Saving $s_{jl} \in S_2$.

Schritt 14: Gilt $s_{jl} = -M_\infty$?

nein Gehe zu Schritt 15.

ja Es kann kein weiteres zulässiges Saving gefunden und damit kann kein Vorgang mehr aus der Menge T_{m+1} den Touren T_1 bis T_m zugeordnet werden. Gehe zu Schritt 16.

Schritt 15: Ist das Saving s_{jl} zulässig?

nein $s_{jl} = -M_\infty$. Gehe zu Schritt 13.

ja Füge den Vorgang v_l zur Tour T_j hinzu. Setze $T_{m+1} = T_{m+1} \backslash v_j$ und entferne alle zugehörigen Savings aus S_2.

Schritt 16: Ermittle alle Nebenvorgänge zu den, in den Touren T_1 bis T_m eingeplanten, Hauptvorgängen und füge sie der Menge NV' hinzu.

Schritt 17: Gilt $NV' = \emptyset$?

nein Wähle ein $v_i \in NV'$. Ermittle alle zulässigen Savings zu den Touren innerhalb der Relativzeiten. Ordne v_i der Tour mit dem größten Saving zu. Setze $NV' = NV' \backslash v_i$. Gehe zu Schritt 17.

ja Alle Nebenvorgänge wurden eingeordnet. Eine Lösung ist mit den Touren T_1 bis T_m gegeben. Die Menge T_{m+1} enthält alle Vorgänge, die noch keiner Tour zugeordnet wurden und die Menge N alle Vorgänge, die nicht automatisch eingeordnet werden konnten.

Der modifizierte Savings-Algorithmus lässt sich in sechs Abschnitte teilen. Bis zu Schritt 2 dient er der Initialisierung und Vorbereitung. Alle weiteren Schritte lassen sich wie folgt gruppieren:

Gruppen	Aufgabe
Schritt 2 - 4	Ordnet zunächst die Terminvorgänge den Touren zu und füllt die Lücken zwischen zwei Terminvorgänge innerhalb einer Tour.
Schritt 5 - 6	Weißt den noch leeren Touren einen Referenzvorgang zu.
Schritt 7 - 11	Ordnet die zeitkritischen Vorgänge den Touren nach ihren maximalen Savings zu.
Schritt 12 - 15	Ordnet nun aus der Menge T_{m+1} den Touren T_1 bis T_m solange Vorgänge zu, bis alle Touren keinen Vorgang mehr aufnehmen können.
Schritt 16-17	Für alle zugeordneten Hauptvorgänge werden die Nebenvorgänge herausgefiltert und innerhalb der Relativzeit den Touren zugeordnet.

Die somit ermittelte Lösung für die Tagestouren ist im Bezug zum linearen Modell nicht zulässig, wenn $N \neq \emptyset$ gilt. Ursache dafür ist ein zu großes Vorgangsvolumen innerhalb eines kurzen Ausführungszeitraums, d. h. nicht alle Vorgänge können vom Monteur innerhalb des Ausführungszeitraumes abgearbeitet werden. Die Teilmenge N der Vorgangsmenge wird dem Dispatcher weitergeleitet, damit dieser diese Vorgänge manuell einplant bzw. entscheidet, was mit diesen Vorgängen passiert.

Die Menge T_{m+1} enthält die Vorgänge, welche bisher keiner Tour T_1 bis T_m zugeordnet wurden. Sie dienen als Ausgangspunkt für eine neue Auftragszuordnung und Planung der darauf folgenden m Plantage.

Im Anschluss kann die Lösung des modifizierten Savings-Algorithmus innerhalb der Touren durch ein exaktes Verfahren, wie zum Beispiel das Branch&Bound-Verfahren für Rundreiseprobleme, verbessert werden. Dies ist möglich, da die Anzahl der Vorgänge pro Tag gering ist. Tourenübergreifend kann diese Verbesserung mit Hilfe des 2 bzw. 3-opt-Verfahrens geschehen. Im Vergleich zu dem Verfahren, welches im Theorieteil 4.4.7 beschrieben wurde, ändert sich nur die Zulässigkeit für den Vertausch von Kanten. Da die viele Bedingungen durch die zeitlichen Vorgaben hinzukommen, ist die Zulässigkeitsprüfung sehr aufwendig und die Möglichkeit der Verbesserung stark eingeschränkt. Die zeitlichen Restriktionen verhindern eine beliebige Verschiebung der Vorgänge wie im klassischen Verbesserungsverfahren.

Die größte Einschränkung entsteht durch die zeitliche Vorgabe der Terminvorgänge, da diese tages- und uhrzeitfix der Optimierung übergeben werden. Durch die uhrzeitfixe Vorgabe der Terminvorgänge, welche einen hohen Anteil bei den Vorgängen bilden (vgl. [34]), ist es bereits schwierig eine zulässige Lösung zu finden. Von Optimierung kann an dieser Stelle kaum noch geredet werden. Ein tatsächliches Optimierungspotenzial wäre vorhanden, wenn die Terminvorgänge, wie bei den Vorgängen mit allgemeiner Terminvorgabe, zunächst nur tagesfix, angegeben werden und nach der Optimierung einer Uhrzeit zugeordnet bekommen. Somit wäre das Problem der Einhaltung der Tageszeiten gegenstandslos und alle Uhrzeiten bräuchten erst nach der Optimierung durch einfache Addition von Fahrzeiten und Plandauern ermittelt werden. Im Modell (TT_2) würden der Parameter u_i und die Nebenbedingungen (5.28i) und (5.28j) entfallen. Das modifizierte Savings-Verfahren würde sich vereinfachen und beim Verbesserungsverfahren wäre ebenso eine größere Flexibilität in dem Vertauschen von Kanten gegeben. Das Problem der Lückenschließung würde reduziert werden, jedoch nicht komplett entfallen. Weiterhin existieren Schaltaufträge bei denen der Hauptvorgang von Fremdfirmen übernommen wird, jedoch die Schaltungen der Monteur erledigt. Somit entsteht eine Lücke zwischen der Hin- und Rückschaltung, in der der Monteur nicht vor Ort sein muss, welche durch einen anderen Vorgang gefüllt werden soll.

5.4 Verallgemeinerungen und Zusammenführung beider Verfahren

5.4.1 Modellerweiterungen

Bisher wurde sowohl die Auftragszuordnung als auch die Tagestouren nur für vereinfachte Probleme ohne Ausnahmen betrachtet. Das praktische Problem seitens envia NSG ist weit aus komplexer, wie im Kapitel 2 deutlich wurde. In diesem Abschnitt wird auf einige Spezialfälle genauer eingegangen und mögliche Änderungen im Modell der Auftragszuordnung bzw. Tagestouren und somit auch die Auswirkungen auf die Lösungsverfahren beschrieben. Dabei werden die Änderungen nur angedeutet und nicht im Detail erläutert. Das Ziel dieses Abschnittes ist es, zu zeigen, dass diese Spezialfälle mit im Modell berücksichtigt werden können. Zudem können nicht alle Spezialfälle behandelt werden, da dies den Rahmen der Masterarbeit sprengen würde und viele Details erst bei einer Umsetzung zu klären sind.

Aufträge mit Bedarf an mehreren Monteuren

In der Praxis sind Aufträge vorhanden, deren Bedarf an Monteuren pro Auftrag größer als eins ist. In der Auftragszuordnung ist nun nicht mehr eine eindeutige Zuordnung eines Auftrags zu einem Monteur gefordert. Dazu wird die erste Nebenbedingung in (AZ) modifiziert:

$$\sum_{j=1}^{|N|} x_{ij} = a_i \quad i = 1 \dots |M| \tag{5.29}$$

mit a_i gleich der Anzahl der benötigten Monteure für den Auftrag m_i. Somit ergibt sich jedoch ein verändertes Modell und die Algorithmen müssen angepasst werden. Bei der Lösungsmethode AZ_a kann der Algorithmus des verbesserten Regret-Verfahren nicht mehr angewendet werden, da durch die Modifikation der ersten Nebenbedingung kein verallgemeinertes Zuordnungsproblem der Form (4.13) vorliegt.

Eine weitere Möglichkeit ist solch einen Auftrag so oft zu kopieren, wie er Mitarbeiter bedarf. Nun muss das Modell jedoch um die Bedingungen erweitert werden, dass ein Auftrag und seine Kopie nicht dem selben Mitarbeiter zugeordnet werden darf. Dies kann innerhalb der Verfahren durch Anpassung der Kosten c_{ij} geschehen.

Bei der Alternative AZ_b sind in den Lösungsalgorithmen ebenso Veränderungen vorzunehmen, jedoch sind diese geringfügig. Zunächst wird auch hier die erste Nebenbedingung im allgemeinen Transportproblem durch (5.29) ersetzt. Eine Überführung zum klassischen Transportproblem und damit die effiziente Lösung ist auch hier möglich. Bei dem Branch&Bound-Verfahren wird der Schritt 2 angepasst. Dabei werden nicht sofort bei dem Zweig $x_{kl} = 1$ alle anderen x_{jl} für $j \neq k$ gleich Null gesetzt. Sondern es muss vorher eine Prüfung statt finden, wie viele Monteure dem Auftrag zugeordnet werden müssen und nur wenn diese Anzahl a_i der Anzahl der Zuordnungen entspricht, können alle weiteren x_{jl} gleich Null gesetzt werden.

Bei den Tagestouren ist nun darauf zu achten, das die zugehörigen Vorgänge zu den Mehrmitarbeitervorgängen auf den selben Tag mit der selben Uhrzeit gelegt werden. Dabei ist es schwer möglich dies für beide zu optimieren. Aus diesem Grund, könnte zunächst eine Optimierung ohne weitere

Sonderstellung dieser Vorgänge für einen Monteur vorgenommen werden. Im Anschluss werden die Vorgänge, welche noch weitere Monteure benötigten als Terminvorgänge festgelegt. Wurde nun einem weiteren Monteur dieser Vorgang zugeordnet, wird dieser bei der Tagestourenplanung nicht mehr verschoben sondern als fest angenommen.

Bearbeitung eines Auftrag von verschiedenen Mitarbeitern

Prinzipiell ist es erwünscht, dass ein Auftrag mit allen seinen Vorgängen von dem gleichen Monteur abgearbeitet wird. Jedoch existieren auch Aufträge, dessen Vorgänge von verschiedenen Monteuren bearbeitet werden sollen. Ist dies von vornherein bekannt, so wird der Auftrag für die Optimierung in zwei Teilaufträge mit den entsprechenden Vorgängen unterteilt und mit Hilfe der Kosten c_{ij} in der Auftragszuordnung kann dann verhindert werden, dass dieser Vorgänge dem gleichen Monteur zugeordnet werden.

Mehrtageseinsätze

Neben *Mehrmitarbeitereinsätze* gibt es ebenso Mehrtageseinsätze. Dies sind Aufträge mit einer Plandauer größer A Stunden, d. h. diese Aufträge können nicht an einem Tag abgearbeitet werden. Bei der Auftragszuordnung spielt dies keine Rolle. Jedoch ist eine Anpassung bei den Tagestouren für diese Art von Aufträgen bzw. Vorgängen notwendig.

Eine Möglichkeit ist die Partitionierung dieser Vorgänge in einzelne Teilvorgängen mit Plandauern kleiner als neun Stunden wie z. B. $2\,h$ vorzunehmen. Nach der Partitionierung können diese dann automatisch eingeplant werden. Um zu verhindern, dass der gesamte Vorgang über einen langen Zeitraum gestreckt wird, d. h. der letzte Teilvorgang eines Vorganges sehr viel später als der erste Teilvorgang eingeplant wird, kann nach Einplanung des ersten Teilvorgangs der Ausführungszeitraum der weiteren Teilvorgänge angepasst werden. Die einzelnen Teilvorgänge gehören zusammen und können ähnlich wie die Nebenvorgänge betrachtet werden. Jedoch ist hier möglich und sogar erwünscht einzelne Teilvorgänge zu einer Tour zuzuordnen. Die Partitionierung soll gewährleisten, dass es möglich ist zwischen dem Mehrtageseinsatz auch Terminvorgänge oder andere Vorgänge höherer Priorität einzuplanen.

Urlaub/Ausfall eines Monteurs

Alle Monteure besitzen im Allgemeinen nicht die gleiche Arbeitszeit, in der sie zur Verfügung stehen. So kann zum Beispiel Urlaub oder Krankheit diese heruntersetzen. Statt ein b für alle Monteure in (AZ) kann mit Hilfe des Parameters b_j dies in der Auftragszuordnung variable gestaltet werden. Der Parameter b_j entspricht der freien Arbeitszeitkapazität des Monteurs n_j und kann zum Beispiel bei Urlaub eines Monteure im betrachtete Zeitraum verringert werden. Es ist dabei darauf zu achten, dass die Summe der Arbeitszeitkapazitäten aller Monteure gleich der Summe der Plandauern ist, d. h.:

$$\sum_{j=1}^{|N|} b_j = \sum_{i=1}^{|M|} p_{m_i} \; . \tag{5.30}$$

Bei den Tagestouren werden dann vor allem die aktuellen Begebenheiten wie Krankheit berücksichtigt. Fällt ein Monteur aus diversen Gründen aus, so findet keine Tourenplanung für diejenigen Tage statt.

Bei kurzfristigen Ausfall können die bereits geplanten Touren der betreffenden Tage nicht erledigt werden. Die zugeordneten Vorgänge dieser Tagestouren werden der Menge der Auftragszuordnung zugewiesen und damit in die nächste Planungsperiode verschoben. Sind darunter jedoch Vorgänge, welche in der Ausfallzeit abgearbeitet werden müssen, wie zum Beispiel Terminvorgänge, muss der Dispatcher eingreifen bzw. ein weiteres Verfahren zur kurzfristigen Planung entwickelt werden.

Störungen

Störungen sind Aufträge, welche sehr kurzfristig den Dispatcher erreichen. Dieser muss dann schnell reagieren und einen Monteur zur Behebung der Störung wählen, welcher eine zu unterbrechenden Auftrag besitzt und in der Nähe der Störung ist. Eine lange Vorplanung ist hier nicht möglich. Fährt ein Monteur zur Störung, so kann er die ihm zugewiesen Vorgänge für diesen Tag nicht alle abarbeiten. Diese nicht abgearbeiteten Vorgänge werden wieder zur Auftragsmenge der Auftragszuordnung gegeben, damit sie zu einem späteren Zeitpunkt eingeplant werden.

Überschneidung von Terminaufträgen

Bei der Auftragszuordnung werden Zeiten in keiner Weise beachtet. Dies kann dazu führen, dass Terminaufträgen mit sich überschneidenden Zeitfenstern einem Monteur zugeordnet werden. Bei den Tagestouren kann dann keine zulässige Lösung angegeben werden.

Die Wahrscheinlichkeit, dass dieser Fall Eintritt, ist sehr gering. Eine einfache Möglichkeit dabei ist, nach der Auftragszuordnung ein Test für jeden Monteur durchzuführen, ob sich Terminaufträge überschneiden. Tritt dieser Fall ein, wird einer der beiden Terminaufträge einem anderen Monteur, welcher dem Ausführungsort am nächsten ist zugewiesen. Bei einer hohen Anzahl an Aufträgen ist dies möglich ohne das Gleichgewicht stark zu beeinflussen. Als Ausgleich kann auch ein Auftrag, welcher kein Terminauftrag ist, mit ähnlicher Plandauer als Tauschobjekt dienen.

Gewichtung der Aufträge/Vorgänge

Die Aufträge bzw. zugehörigen Vorgänge besitzen zum Teil unterschiedliche Prioritäten. So sind zum Beispiel Terminvorgänge unbedingt einzuhalten. Des Weiteren sind die Wichtung der einzelnen Vorgänge unterschiedlich, so sind beispielsweise Vogelschutzmaßnahmen wichtiger als Inspektionen. Da in manchen Fällen das Vorgangsvolumen größer als das Arbeitsvolumen ist, können in diesem Fall nicht alle Vorgänge einer Tour zugeordnet werden (siehe Abschnitt 5.3.3). Damit jedoch die Vorgänge bei der Zuordnung zu den Touren bevorzugt werden, welche eine höher Gewichtung besitzen, wird eine Gewichtungsfunktion[11] $g(i)$ zur Zielfunktion hinzugenommen.

[11] Diese kann ähnlich interpretiert werden wie der Auftragsindex bei der Auftragseinplanung aus Kapitel 3

In der Zielfunktion ergibt sich ein weiterer Term und die neue Zielfunktion lautet somit:

$$\text{ZF"}: \quad \sum_{k=1}^{m} \sum_{i=0}^{|V|} \Big[\sum_{j=0}^{|V|}(c_{ij} + \alpha\, l_{ij})x_{ijk}\Big] - g(i) \cdot y_{ik} \rightarrow min \tag{5.31}$$

Im modifizierten Savings-Verfahren wird die Gewichtungsfunktion bei der Berechnung der Savings mit berücksichtigt. Somit wird ein Vorgang bevorzugt, der neben der Entfernung auch eine höhere Gewichtung besitzt. Dies bedeutet im Fall, dass zwei Vorgänge, welche eine ähnliche Entfernung besitzen, der Vorgang bevorzugt wird, welcher ein höheres Gewicht besitzt.

Ziel: Minimaler Ressourcenverbrauch

Der minimale Ressourcenverbrauch kann nur in der Auftragszuordnung berücksichtigt werden. Auch hier kann die Zielfunktion so modifiziert, dass neben der Entfernung ebenso der Ressourcenverbrauch eine Rolle spielt. Zu den Kosten c_{ij} wird der Ressourcenverbrauch addiert. Dazu ist es jedoch notwendig ein geeignetes Maß zu finden. Dies soll an dieser Stelle nicht definiert werden.

Ziel: Zeitnahes Abarbeiten von Inspektionen in einem Ortsteil

Das zeitnahe Abarbeiten von Inspektionen in einem Ortsteil ist ein Problem der Tagestouren und kann ähnlich wie nach der Partitionierung der Mehrtageseinsätze über den Ausführungzeitraum geschehen. Dazu wird nach Einplanung der ersten Inspektion in einen Ortsteil der Ausführungzeitraum der weiteren Inspektionen in diesem Ortsteils verringert.

Eine weitere Möglichkeit ist nach Einplanung einer Inspektion das Anpassen der Gewichtsfunktion $g(i)$ der weiteren Inspektionen v_i in diesem Ortsteil vorzunehmen. Dazu wird das Gewicht dieser Inspektionen vergrößert, so dass eine baldige Einplanung durch die Optimierung bevorzugt wird.

Berücksichtigung von Schaltaufträgen

Wie bereits im Kapitel 2 beschrieben, unterstehen die Schaltaufträge einer besonderen Stellung, da sie eine Stromunterbrechung für die betreffenden Kunden mit sich ziehen. Damit in einem Netzobjekt möglichst nicht ständig eine Schaltung durchgeführt wird, ist es notwendig, dass innerhalb des gleichen Netzobjektes Schaltaufträge zusammengefasst werden. In der Auftragszuordnung kann dies im voraus geschehen. Dabei werden diese zu einem Paket zusammengefasst und einem Monteur zugeordnet.

Bei den Tagestouren werden dann bei einer Tour mit bereits zugeordnetem Schaltvorgang die Vorgänge bevorzugt, welche ebenso Schaltungen im selben Netzobjekt sind. Dies kann durch Einführung eines weiteren Schrittes im Algorithmus geschehen. Bei jeder Zuordnung eines Vorganges zu einer Tour wird getestet, ob dieser eine Schaltung beinhaltet und weitere Vorgänge mit Schaltungen im selben Netzobjekt existieren. Ist dies der Fall, wird ein solcher Vorgang zur Tour hinzugenommen.

5.4.2 Gesamtverfahren

Beim alternativen Ansatz wurde in Auftragszuordnung und Tagestouren unterschieden. Während der Auftragszuordnung werden alle Aufträge gleichmäßig und unter Berücksichtigung der Ressourcen den Monteuren zugewiesen. Bei der Zuweisung wird die Gesamtentfernung zwischen Ausführungsort und Startadresse des Monteurs minimiert. Zur Lösung der Auftragszuordnung wurden zwei verschiedene Algorithmen angegeben. Welche dieser beiden Methoden zu bevorzugen ist, konnte nicht herausgestellt werden. Die Umsetzung und der Vergleich beider Methoden könnte Aufgabe einer weiteren Arbeit sein.

Im Anschluss zu der Auftragszuordnung findet die Tagestourenplanung statt. Dabei werden unter Beachtung aller zeitlichen Restriktionen, wie z. B. Ausführungszeiträume, jedem einzelnen Monteur Tagestouren für die m Plantage zugewiesen. Mit Hilfe des modifizieren Savings-Verfahren und des 2 bzw. 3-opt-Verfahrens können die Touren hinsichtlich der Entfernungen der Aufträge optimiert werden, was einer Minimierung der Fahrzeit nach sich zieht.

Bei der Auftragszuordnung werden die Aufträge für die kommenden drei Monate den Monteuren zugewiesen. Aus dieser zugewiesenen Auftragsmenge eines Monteurs werden dann die Tagestouren für z. B. zwei Arbeitswochen eine Woche im Voraus erstellt. Es ist dabei nicht möglich bereits vorhandene Touren aufzugreifen und bei Änderung, beispielsweise beim Hinzukommen neuer Terminaufträge, diese in die Touren einzubinden. Aus diesem Grund sind beide Verfahren in der Planungsphase anzusiedeln. Für die Berücksichtigung aktueller Änderungen in bereits geplanten Touren ist es notwendig ein weiteres Verfahren zu entwickeln. Auf die Entwicklung solch eines Verfahrens wurde innerhalb dieser Arbeit verzichtet.

Am Anfang der Planung steht die Optimierungsaufgabe der Auftragszuordnung. Im Anschluss zur Auftragszuordnung werden $|N|$ Optimierungsaufgaben, die Tagestouren, gelöst. Die Auftragszuordnung liefert die Menge der Aufträgen M_j für jeden einzelnen Monteur j, welche die Basis für die Tagestouren bildet. Diese Auftragsmenge bzw. Vorgangsmenge wird bei den Tagestouren wiederum in $k + 1$ Mengen zerlegt. Die ersten k Mengen spiegeln die Tagestouren der Planungstage wider und die $k + 1$. Menge enthält alle Vorgänge, welche nicht zu den k Planungstagen zugeordnet werden konnten. Die Vorgänge der $k + 1$. Menge werden der Auftragszuordnung wieder zurückgeben, damit die zugehörigen Aufträge in der nächsten Planungsphase wieder mit berücksichtigt werden können. Das Zusammenspiel von Auftragszuordnung und Tagestouren ist in der Abbildung 5.10 graphisch dargestellt.

Im vorangegangenen Abschnitt wurde kurz angeschnitten wie die Modelle und Algorithmen hinsichtlich der Spezialfällen erweitert werden können. Bei allen Verfahren ist zu beachten, dass keinerlei Schnittstellen definiert wurden, d. h. zum Beispiel, dass eine Zuweisung eines Auftrages zum Monteur in der Auftragszuordnung beinhaltet, das nur die zugehörigen Vorgänge mit allen benötigten Informationen zum Auftrag dem Monteur zugewiesen werden.

Das Ziel dieses Kapitels war es, eine alternative Herangehensweise zum Ansatz aus Kapitel 3 zu diskutieren, die Modelle aufzustellen und Lösungsverfahren aufzuzeigen. Es wurden Modelle geschaffen, welche das Planungsproblem von envia NSG recht gut widerspiegeln und welche mit bekannten Methoden effektiv zu lösen sind. Eine algorithmische Umsetzung der Auftragszuordnung und Tagestouren ist damit möglich. Alle beschrieben Verfahren und Algorithmen sind effizient und liefern voraussichtlich in angemessener Zeit eine gute Lösung.

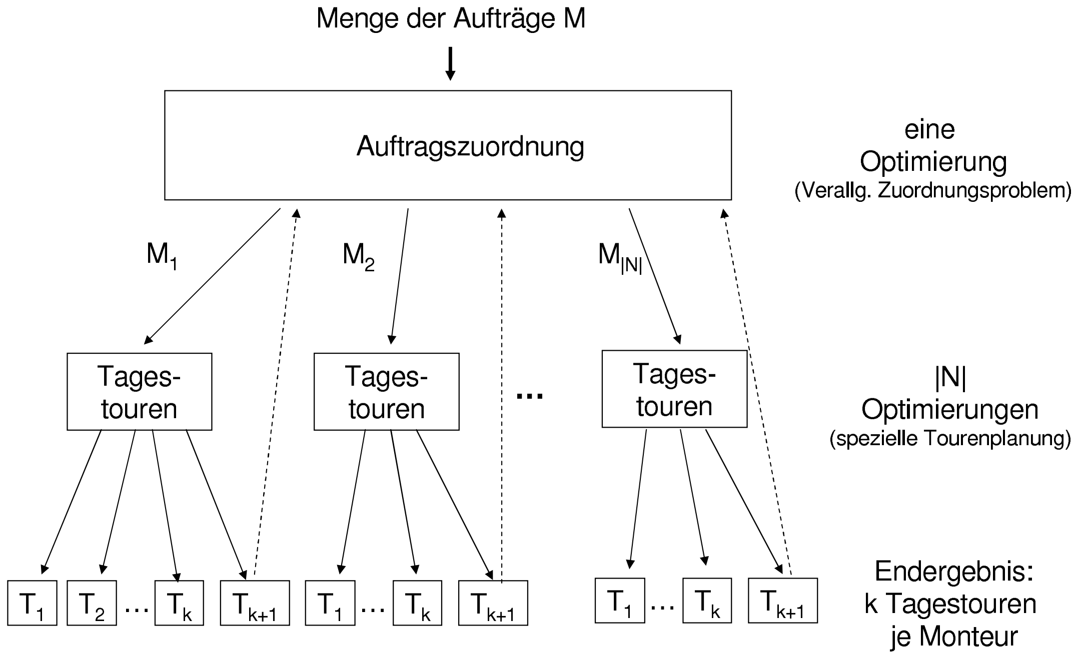


Abbildung 5.10: Gesamtverfahren

6 Zusammenfassung und Ausblick

Im ersten Teil der Arbeit wurde eine Problemstellung der Firma envia NSG vorgestellt. Die Dispatcher sollen bei ihrer täglichen Arbeit mit Hilfe einer Optimierungssoftware unterstützt werden. Dabei werden eine Reihe an Forderungen an die Software gestellt, damit das Ergebnis die Realität so gut wie möglich abbildet. Im zweiten Kapitel wurden zunächst die wichtigsten Begriffe erläutert, um dann im Kapitel 3 ein genaue Beschreibung der betrieblichen Ziele vorzunehmen. Es wurde ersichtlich, dass das Problem sehr komplex ist und viele Besonderheiten beinhaltet.

Die Entwicklung und Umsetzung eines Lösungsverfahrens wurde der Firma F/L/S übertragen. Aus diesem Grund wurde die Überführung der betrieblichen Ziele in Optimierungsziele und die Beschreibung der Nebenbedingungen unter dieser Voraussetzung vorgenommen. Das Verfahren Qualicision® von F/L/S beruht auf der Fuzzy Logik und ist in der Lage Polyoptimierungsprobleme in angemessener Zeit zu lösen. Somit ist es möglich, das Problem von envia NSG als Optimierungsproblem mit mehreren Zielfunktionen zu beschreiben und die Nebenbedingungen weich zu formulieren. Die Einplanung der Aufträge bzw. Vorgänge geschieht hier in zwei Schritten zuerst wird die *Auftragseinplanung* und im Anschluss die *Vordisposition* durchgeführt. Bei der Auftragseinplanung werden die Aufträge auf die Tage verteilt und in der Vordisposition dann den Einheiten zugewiesen, sowie die Reihenfolge der Abarbeitung für jeden einzelnen Tag angegeben. Für beide Aufgaben wurden entsprechende Optimierungsziele formuliert. Die mathematische Modellierung und Entwicklung der Algorithmen wird der Firma F/L/S überlassen.

Im zweiten Teil der Arbeit wurde ein alternatives Verfahren entwickelt. Auch hier findet eine Teilung des Problems statt. Es wird zwischen *Auftragszuordnung* und *Tagestouren* unterschieden. Dabei werden nun in der Auftragszuordnung zunächst die Aufträge den Einheiten zugeordnet und in den Tagestouren dann die Vorgänge den Tagen zugeordnet, sowie die Reihenfolge der Abarbeitung festgelegt. Die zugehörigen Modelle beruhen auf der klassischen Optimierung mit einem Optimierungsziel und harten Nebenbedingungen. Die Auftragszuordnung wurde als verallgemeinertes Zuordnungsproblem modelliert und die Tagestouren als spezielles Tourenproblem. Die Grundgerüste der Algorithmen beziehen sich zum Teil auf in der Literatur bekannte Verfahren. Zur Anpassung an das spezielle Optimierungsproblem von envia NSG wurden neue Ideen entwickelt, damit eine effiziente Lösung möglich ist. Auf die Umsetzung der Verfahren wurde verzichtet.

Ein Vergleich beider Herangehensweisen ist theoretisch nicht möglich. Nur nach Umsetzung und Anwendung beider Methoden auf die gleiche Datenbasis ist eine Gegenüberstellung denkbar. Jedoch ist auch dann eine Beurteilung schwierig, da das Optimierungsproblem nicht *die* optimale Lösung besitzt. Zur Bewertung müssten verschiedene Kriterien herangezogen werden. Eines der wichtigsten Kriterien ist die Praxistauglichkeit der Lösung, d. h. wie gut wurde das reale Problem widergespiegelt und wie gut unterstützt die Lösung den Dispatcher. Ein Maß dafür könnte die Anzahl der Aufträge bzw. Vorgänge sein, welcher der Dispachter manuell neu einplanen muss, weil das Ergebnis der Optimierung nicht umsetzbar wäre (z. B. durch Verstoß gegen eine Bedingung). Bei der Beurteilung spielt neben der Praxistauglichkeit auch die Kostenminimierung eine Rolle, also zum Beispiel die Höhe der durchschnittlichen Fahrzeiten oder die durchschnittliche Dauer der nicht belegten Arbeitszeit zwischen zwei Terminvorgängen an einem Tag bzw. das Verhältnis von tatsächlicher Arbeitszeit zur Einsatzzeit, welche sich aus Arbeitszeit, Fahrzeit und Wartezeit zusammensetzt.

1. Resultat Es existiert nicht *die* optimale Lösung. Bei der Polyoptimierung werden in der Regel eine Menge von Kompromisslösungen und bei den Näherungsverfahren der klassischen Optimierung suboptimale Lösungen, d. h. fast optimale Lösungen, erhalten. Die Güte der Lösungen sollte anhand von Praxistauglichkeit und Kosteneinsparungen gemessen werden.

Die im Kapitel 5 beschriebene alternative Herangehensweise ist unabhängig von der den Algorithmen zugrunde gelegten Mathematik, d. h. auch mit Hilfe der Fuzzy-Optimierung können Modelle dieser Herangehensweise gelöst werden. Der Vorteil der alternativen Herangehensweise ist, dass bereits einige der Optimierungsziele und Nebenbedingungen, welche zusätzlich beim ersten Ansatz aufgestellt wurden, automatisch erfüllt sind. Zum Beispiel müssen keine Auftragspakete erstellt werden, welche gleichverteilt im Gebiet liegen sollen. Zudem werden die Ressourcen von Beginn an beachtet und es kommt später zu keinem Konflikt bei der Zuordnung der Vorgänge zu den Monteuren. Der Ablauf der alternativen Herangehensweise unterscheidet sich von der des Dispatchers, d. h. erst werden die Aufträge einer gesamten Planungsperiode den Einheiten zugeordnet und dann findet die Tagestourenplanung statt. Dies behindert jedoch die Arbeit des Dispatchers nicht, da dieser nur das Endergebnis, d. h. den Vorschlag eines Einsatzplanes für jeden Tag einer festgelegten Planungsperiode, erhält und der konkrete Ablauf der Verfahren nicht interessant für den Dispatcher ist. Die zwei Teiloptimierungsaufgaben des alternativen Ansatzes enthalten jeweils nur eine Zielfunktion. Optimierungsprobleme mit nur einer Zielfunktion sind im Allgemeinen schneller zu lösen.

2. Resulat Die Zuordnung der Aufträge *zuerst* zu den Einheiten und dann zu den Tagestouren ist günstiger, da damit der Einsatz der Polyoptimierung umgangen werden kann. Der Ansatz ist unabhängig davon, welche Mathematik den Algorithmen zu Grunde gelegt wird, d. h. egal ob Fuzzy- oder klassische Optimierung.

Es ist zudem möglich aus dem Ergebnis der Optimierung Erkenntnisse über das Ausgangsproblem zu erlangen. Bei den Tagestouren ist dies zum Beispiel die Menge N. Diese beinhaltet alle die Vorgänge, welche nicht innerhalb ihres Ausführungszeitraums einer Tour zugeordnet werden konnten und dem Dispatcher zur manuellen Bearbeitung übergeben werden. Ist die Mächtigkeit dieser Menge sehr groß, so ist die Arbeitszeitkapazität der Einheiten weit aus geringer als das Vorgangsvolumen.

Des Weiteren können aus der Optimierung auch Erkenntnisse über das praktische Problem gewonnen werden. In der Optimierung gilt im Allgemeinen, je weniger Nebenbedingungen im Modell vorhanden sind, desto geringer ist der Speicherbedarf, die Rechenzeit und desto höher das Optimierungspotential. Viele Nebenbedingungen schränken die Suche nach einer gute Lösung stark ein und es kann sogar dazuführen, das keine zulässige Lösung gefunden wird. Um ein Problem so genau wie möglich zu modellieren, ist es zwar notwendig dieses mit Hilfe von verschiedenen Nebenbedingungen zu beschreiben, jedoch gilt auch hier: *„Weniger ist manchmal mehr“*. Jede Nebenbedingung sollte auf ihren Sinn und den Nutzen untersucht werden. Dies kann auch im Nachhinein, also innerhalb der Testphase geschehen. Es ist ratsam im Test die Grenzen der Nebenbedingungen zu variieren oder sogar einige Nebenbedingungen auszuschalten.

3. Resultat Innerhalb der Testphase sollte eine Untersuchung des Nutzens jeder Nebenbedingungen durchgeführt werden. Überflüssige Nebenbedingungen sollten entfernt werden, um einerseits das Optimierungspotenzial zu erhöhen und anderseits die Rechenlaufzeit zu verkürzen.

Um ein hohes Optimierungspotential und damit eine nützliche Lösung für das Problem zu erlangen, ist es oftmals auch notwendig nicht nur die Optimierung dem Prozess, sondern auch den Prozess der Optimierung anzugleichen. Markant hierfür sind die Terminvorgänge mit konkreter Uhrzeitvorgabe. Würde zunächst nur der Tag angegeben werden und keine Uhrzeit, so könnte die Optimierung weitaus freier nach einer guten Lösung suchen, die Rechenzeit wäre kürzer und auch der Speicherbedarf geringer, da die Prüfung einiger Nebenbedingungen entfallen würde. Im Abschnitt 5.3 wurde gezeigt, dass sich das Verhältnis von Arbeitszeit und Fahrzeit zugunsten der Arbeitszeit verändern (siehe Beispiel und Abbildung 5.8) und somit Kosten gespart werden würden

4. Resultat Mit der Einführung der automatischen Tourenplanung ist es empfehlenswert, bei der mittelfristigen Planung keine Terminvorgänge mit Uhrzeitangabe vorzugeben, sondern die Uhrzeiten erst nach der Optimierung festzulegen. Der Durchlauf einer ersten Optimierung und damit die konkrete Festlegung der Uhrzeiten kann 2-3 Wochen im Voraus geschehen und damit noch rechtzeitig dem Kunden bzw. der Fremdfirma mitgeteilt werden.

Ausblick

Ziel der Arbeit war die mathematische Aufbereitung und Modellierung des Planungsproblems von envia NSG. Eine rechentechnische Umsetzung innerhalb dieser Arbeit war nicht vorgesehen und hätte den Arbeitsumfang und zeitlichen Rahmen gesprengt. Sollte jedoch die Weiterverfolgung der, in der Arbeit vorgeschlagenen und durch mathematische Algorithmen untersetzten, alternativen Herangehensweise vorgesehen sein, empfehlen sich die folgenden Schritte:

a) Umsetzung beider in Abschnitt 5.2 beschriebener Verfahren in der Auftragszuordnung: Nach der Umsetzung der Lösung mittels Anpassung der rechten Seite und mit Hilfe der Relaxation können Tests entscheiden, welches Verfahren bei der Zuordnung der Aufträge besser geeignet ist. Am Ende sollte sich für ein Verfahren entschieden werden, welches weiter verfolgt wird.

b) Umsetzung des Verfahrens für die Tagestouren: Das als Eröffnungsverfahren vorgestellte modifizierte Savingsverfahren und mit anschließenden 2 bzw. 3-opt-Verfahren als Verbesserungsverfahren sind in einem Programm umzusetzen und zu testen.

c) Integration der Spezialfälle: Nach der Umsetzung der Auftragszuordnung und der Tagestouren sollten die Spezialfälle, welche im Abschnitt 5.4.1 angesprochen wurden, in die Algorithmen integriert werden.

d) Entwicklung eines Algorithmus für kurzfristige Planungen: Die Tagestourenplanung liefert eine unabhängige Lösung, welche jedoch bisherige geplante Touren nicht berücksichtigt. Das Verfahren der Tagestouren kann in der mittelfristigen Planung angesiedelt werden, d. h.

es dient zur Erstplanung von z. B. 10 Arbeitstagen eine Woche im Voraus. Für die Anpassung bereits geplanter Touren an aktuelle Änderungen, wie z. B. kurzfristiger Ausfall eines Mitarbeiters, ist es notwendig ein weiteres Verfahren zu entwickeln und umzusetzen.

e) Definition der Schnittstellen: Zwischen den Daten aus RessMa und den Programmen der alternativen Herangehensweise müssen die Schnittstellen definiert werden.

f) Test mit realen Daten: Am Ende sollten die Verfahren der Auftragszuordnung und Tagestouren auf reale Daten angewendet und eine Bewertung dieser Verfahren vorgenommen werden. Zudem kann dann ein Vergleich der Ergebnisse mit denen der Auftragseinplanung und Vordisposition statt finden.

Für eine Weiterverarbeitung in diesem Sinne wären studentische Projekte sowie Bachelor- und Masterarbeiten denkbar.

Begriffe

Begriffe aus dem Unternehmen

Alle Begriffe in dieser Kategorie werden bezogen auf das Unternehmen enviaM definiert. Die Erläuterungen und Definitionen sind nicht allgemeingültig. Zum Teil wurden die Definitionen aus dem Pflichtenheft [2] entnommen.

Anlagenmanagement (AM)
Anlagenmanagement ist die Organisationseinheit der envia NSG, die als Anlagenbetreiber die Unternehmerpflicht für den sicheren Betrieb und ordnungsgemäßen Zustand der elektrischen Anlagen wahrnimmt. Durch die betriebliche Struktur der envia NSG wird sichergestellt, dass für alle Anlagenteile eindeutig ein Betreiber zugeordnet ist.

Arbeitsvolumen
Das Arbeitsvolumen setzt sich aus der Anzahl der Verfügung stehenden Einheiten ihren Arbeitszeit zusammen.

Auftrag
Ein Auftrag umfasst alle Tätigkeiten zu einer Instandhaltungsmaßnahme oder Störung. Die unterschiedlichen Tätigkeiten eines Auftrags werden in einzelne Vorgänge herunter gebrochen.

Auftragsvolumen
Summe der Plandauer der Einsätze zuzüglich der Fahrzeit und eventuell festgelegter Vor- und Nacharbeitszeiten zum Einsatz.

Ausführungszeitraum
Zeitraum in welchem der Auftrag auszuführen ist. Dieser wird beim Anlegen des Auftrags in Abhängigkeit von der Priorität und vom Auftragstyp festgelegt.

Chef der Einheit
Im Allgemeinen wird jedem Fahrzeug eine verantwortlicher Mitarbeiter zugewiesen. Dieser fährt im Normalfall das Fahrzeug und ist damit Chef der Einheit. Nur in Ausnahmefällen wie zum Beispiel bei Krankheit, kann das Fahrzeug von einem anderen Mitarbeiter bedient werden.

CommBox
Jede Einheit besitzt eine CommBox. Diese empfängt die Daten von der Zentrale über GPRS, speichert diese ab und stellt sie dem Außendienstmitarbeiter zur Verfügung. Ein Monteur erhält zum Beispiel die disponierten Vorgänge der nächsten Tage.

Dispatcher
Ein Dispatcher ist ein Mitarbeiter, welcher für die Disposition verantwortlich ist.

Disposition
Die Disposition beschreibt den Vorgang der Zuordnung von Einsätzen zu Einheiten. Dabei werden

Ressourcen, wie z. B. Qualifikationen, Berechtigungen, Ausrüstungen, Prioritäten und Verfügbarkeiten berücksichtigt.

Einheit
Die Einheit ist der Empfänger von disponierten Einsätzen. Der Einheit können Mitarbeiter und/oder ein Fahrzeug zugeordnet werden. Damit verfügt die Einheit über die Ressourcen der zugeordneten Mitarbeiter und Fahrzeuge (Qualifikationen, Berechtigungen, Ausrüstungen, Prioritäten und Verfügbarkeiten), welche die Einheit zur Ausführung von Einsätzen befähigen. Eine Einheit hat folgende Eigenschaften:

- Eine Einheit hat einen Start- und Endpunkt (z. B. Wohnadresse) an dem der Arbeitstag beginnt und endet.
- Verfügbarkeiten aufgrund von Dienst- und Schichtplänen, sowie Verfügbarkeitsplänen der Fahrzeuge und Materialien/Werkzeuge sind definiert. Der Start- und Endzeitpunkt der Arbeitszeit wird durch die Dienstpläne der zugeordneten Mitarbeiter bestimmt.
- Die Auslastung der Einheit je Tag steigt mit der Anzahl und Dauer der disponierten Einsätze. Dabei darf eine bestimmte Grenze nicht überschritten werden.
- Einer Einheit ist einer Organisationsebene, z.B. MS/NS oder Schutz- und Leittechnik, mit einer Summe von möglichen Einsatzgebieten zugeordnet. Der Einsatz der Einheit ist nicht an die Einsatzgebiete gebunden. Sie dienen lediglich einer sinnvollen Vorauswahl bei der Disposition.

Einsatz
Der Einsatz ist ein, an eine Einheit elektronisch übermittelter Vorgang zur Durchführung eines Arbeitsauftrags. Der Einsatz beschreibt die durchzuführende Tätigkeit und wird auf eine Einheit disponiert. Jeder Einsatz wird genau von einer Einheit bearbeitet werden. Einsätze werden typisiert, d.h. gemeinsame Eigenschaften werden zu Typen gruppiert. Ein Einsatz ist immer einem Einsatztypen zugeordnet. Ein Einsatz hat folgende Eigenschaften:

- Ein Einsatz besitzt einen geographischen Einsatzort
- Einsätze sind termingebunden oder können in einem festgelegten Zeitfenster verschoben werden.
- Termingebundene Einsätze haben einen Ausführungszeitpunkt.
- Einsätze besitzen eine Priorität, welche die Verschiebbarkeit bestimmt.
- Einsätze besitzen einen Status, der den Stand der Abarbeitung anzeigt
- manche Einsätze können unterbrochen werden.
- Einsätze besitzen eine Plandauer. Mit diesen Plandauern belasten sie die Auslastung der Einheiten. Bei Fremdleistung wird die Einheit nicht belastet.
- Einsätze verlangen Ressourcen, wie Qualifikationen, Ausrüstungen und Berechtigungen der bearbeitenden Einheit.

Einsatzdauer
Die Einsatzdauer bei der Einplanung ergibt sich aus der abgeschätzten Fahrzeit zum Einsatz und der Plandauer. Zur Auswertung wird die Einsatzdauer dann mit der tatsächlichen Fahrzeit und der Ist-Dauer angegeben.

Einsatzgebiet
Das Einsatzgebiet ist das kleinste im WFM definierte Gebiet. Das zuständige Gebiet für eine Organisation kann mehrere Einsatzgebiete umfassen.

Fahrzeit
Zeit vom Standort der disponierten Einheit zum Einsatzort. Es gibt grundsätzlich drei Möglichkeiten zur Berücksichtigung der Fahrzeit:

- Pauschalwert pro Einsatz (z.B. 30 min) (Stand heute)
- Nach Luftlinienentfernung
- Nach Routenentfernung (auf Basis von routingfähigem Kartenmaterial)

Fremdleistung
Nicht alle Aufträge werden von den Einheiten der envia NSG bearbeitet. Leistung, welche von Fremdfirmen erledigt werden, zählen unter Fremdleistungen.

Hauptvorgang
Der Vorgang, welcher in direkter Beziehung zum Auftrag steht, wird dabei als Hauptvorgang ausgezeichnet. Alle weiteren Vorgänge sind begleitende, die in zeitlichen Beziehungen zum Hauptvorgang stehen. Diese Beziehung bildet den Start oder das Ende der Einsätze in Abhängigkeit zum Hauptvorgang ab.

Mehrtageseinsätze
Mehrtageseinsätze sind Einsätze, welche eine Plandauer größer als acht Stunden besitzen und somit über mehrere Tage abgearbeitet werden müssen.

Modulares Anlagen-Bewertungs- und Instandhaltungspaket (MABI)
MABI ist ein elektronisches System zur Auftragserzeugung und -abwicklung, zur Störungs- und Ereigniserfassung und zur Vorhaltung von Instandhaltungsdaten und Inspektionsergebnissen.

Optimierungssperre
Die Optimierungssperre ist ein Parameter in der Datenbank. Sie kann manuell durch den Dispatcher gesetzt werden oder automatisiert. Bei gesetzter Optimierungssperre darf der Auftrag nicht automatisch vom System verschoben oder automatisch vordisponiert werden.

Organisation
Innerhalb des Unternehmens eine Gruppe, die für die Umsetzung der Arbeiten in einem zugeordneten Gebiet und Organisationsebene zuständig ist. Einer Organisation sind die Einheiten eindeutig zugeordnet. Sie muss nicht mit den Struktureinheiten der Unternehmensstruktur identisch sein.

Organisationsebene
Die Organisationsebene bestimmt zusammen mit dem Einsatztyp die zuständige Organisation am Einsatzort. Sie ist beim Einsatztyp hinterlegt.

Qualicision
Qualicision® ist eine Technologie der F/L/S Fuzzy Logik Systeme GmbH und arbeitet wie Fuzzy Logik mit Zwischenwerten, erweitert aber den modellierbaren Aussagenbereich der Fuzzy Logik zwischen ´Ja´ (logisch 1) beziehungsweise ´Nein´ (logisch 0) um Aussagen der Form ´Auf keinen Fall (Qualicision -logisch -1) und alle zwischen 0 und -1 liegenden Möglichkeiten von Gegenaussagen. Mit dieser Erweiterung sind neben positiven Bewertungen gleichfalls negative Auswirkungen von Handlungen, etwa Verbote und Warnungen, effizient modellierbar. Mit Qualicision ist die Modellierung von zielorientierten Entscheidungsprozessen möglich. Der Begriff eines Ziels

wird bei Qualicision als ein anzustrebendes Qualitätsmerkmal des Zielzustandes des jeweils modellierten Prozesses verstanden. Mehrere Ziele definieren ein vom Prozess zu erreichendes Ziel-Qualitätsspektrum, das für den modellierten Prozess angestrebt wird.

Qualicision-Optimierung
Optimierung der automatisierten Auftragseinplanung und Vordisposition mittels Qualicision®.

Plandauer
Die Plandauer ist der zu erwartende Arbeitsaufwand pro Einsatztyp in Stunden. Momentan ist das ein angenommener Mittelwert und stellt einen relativ großen Unsicherheitsfaktor dar.

Priorität
Die Priorität bestimmt Dringlichkeit eines Auftrages und Einsatzes. Folgende Prioritäten sind definiert:

- Störung
- Termingebunden (fester Termin mit Uhrzeit)
- Allgemeiner Termin (nur Datum mit Zeitraum)
- Turnus (mit Ausführungszeitraum)
- Nicht priorisiert

Ressourcen
Ressourcen sind Ausrüstungen, Qualifikationen und Berechtigungen, welche einer Einheit zugeordnet sind. Zur Ausführung eines Einsatzes sind bestimmte Soll-Ressourcen erforderlich.

Ressourcen-Management-System (RessMa)
RessMa ist das Auftrags- und Einsatzleitsystem und ein integrierter Bestandteil des WFM-Systems der envia NSG.

Schaltantrag (SAN)
Der Schaltantrag ist der Antrag des für den Betrieb der Anlage Verantwortlichen zum Freischalten bzw. zur Durchführung anderer Schalthandlungen, zur Inbetrieb- und Außerbetriebnahme eines elektrischen Betriebsmittels bzw. einer Anlage, zum Arbeiten an Schutz-, Steuer- und Messeinrichtungen oder der Durchführung der Arbeiten unter Spannung (AuS-Mittelspannung) bei der zuständigen Schaltbefehlsstelle.

Schaltauftrag
Schaltauftrag ist ein von der zuständigen Schaltbefehlsstelle zur Ausführung bestätigter Schaltantrag

Schalteinsatz
Schalteinsatz ist ein Sonderfall eines Einsatzes zur Ausführung von Schalthandlungen und ist der Oberbegriff für alle Einsatztypen, welche Schalthandlungen beinhalten.

Sondervorgang/Sondereinsatz
Bei den Sondervorgängen werden mehr als ein Mitarbeiter zur Erledigung dieser vorausgesetzt, d. h. zeitgleich befinden sich mehrere Einheiten im selben Ortsteil. Sondereinsätze sind disponierte Sondervorgänge.

Tablet-PC
Ein Tablet-PC ist ein tragbarer stiftbedienbarer Computer. Die Eingabe erfolgt direkt mit einem Stift oder dem Finger auf dem Bildschirm.

Workforcemanagement (WFM), allgemein
Unter Workforcemanagement versteht man die optimierte Disposition von Ressourcen (z. B. Mitarbeiter, Fahrzeuge, Maschinen, etc.), wobei entscheidend ist, dass jeweils die richtige Ressource zur richtigen Zeit entsprechend dem Auftrag disponiert wird. Wenn es sich bei der Ressource um Außendienstmitarbeiter handelt, gehört auch deren mobile Anbindung zum Workforce Management. Das Workforce Management berücksichtigt bei der Einteilung der Servicemitarbeiter deren Qualifikationen, mitgeführtes Material und deren Auslastung. Es wird permanent in Echtzeit geplant, wobei die Daten der Servicemitarbeiter automatisch ins Backoffice übertragen werden und somit kurzfristige Änderungen schnell berücksichtigt werden können [4].

Workforce-Management innerhalb der enviaM
Workforce-Management dient zur Unterstützung der planbaren und nicht planbaren Aktivitäten bei Baumaßnahmen, Inbetriebnahmen, Betrieb, Instandhaltung und Störungsbeseitigung in Versorgungsnetzen zur kostenoptimalen Ressourcensteuerung (Personal, Werkzeug und Material) und Ergebnisdokumentation. Integriertes WFM bildet darüber hinaus die Abrechnungsebene ab.

Workforce-Management-System (WFM-System)
Workforce Management Systeme sind intelligente Systeme, die permanent, also in Echtzeit, aktuelle Änderungen berücksichtigen können. Der Kontakt zwischen Mitarbeitern und dem System kann auf vielfältige Weise hergestellt werden, z. B. über WAP, Internet oder Telefon [5].

Verfügbares Arbeitsvolumen

a) der Organisation: Summe des verfügbaren Arbeitsvolumens aller zugeordneten Mitarbeiter.

b) der Einheit: 8-10 h pro Tag und Mitarbeiter unter Berücksichtigung der Gleitzeit, max. 38 h in der Woche. Einer Einheit können auch mehrere Mitarbeiter gemäß Zuordnungsplan zugeordnet sein. Das verfügbare Arbeitsvolumen skaliert nicht mit der Anzahl der zugeordneten Mitarbeiter. Bei der Einplanung einer Einheit verringert sich das Arbeitsvolumen aller zugeordneten Mitarbeiter um die Sollarbeitszeit des zugeordneten Einsatzes.

Verfügbarkeit
Die Verfügbarkeit einer Einheit, eines Mitarbeiters oder einer Ressource leitet sich aus dem Dienstplan oder dem Verfügbarkeitsplan der Ressource ab. Die Verfügbarkeiten werden viertelstündlich geführt.

Versorgungsgebiet der enviaM Das Versorgungsgebiet der enviaM erstreckst sich von West-Sachsen über Süd-Brandenburg, ein kleiner Teil Sachsen-Anhalts und Thüringens.

Vordisposition
Wie Disposition, aber keine Einsatzübermittlung an die Einheiten.

Vorgang
Vorgang ist eine Aktivität zur Ausführung eines Auftrages, welche andere Ressourcenanforderungen oder andere Einsatz- bzw. Termininformationen besitzt. Der Vorgang, welcher unmittelbar die Ausführung des Auftrages beinhaltet ist der Hauptvorgang. Alle anderen sind begleitende Vorgänge, wie der Schaltantrag, die Schalteinsätze und weitere begleitende Einsätze, sowie Benachrichtigungen zur Informationsübermittlung.

Abkürzungen und Notationen

Abkürzungen

BZ	betriebliches Ziel/Betriebsziel
F&E	Forschung und Enwicklung
HS/MS/NS	Hoch-, Mittel- und Niederspannung
OZ	Optimierungsziel
SAN	Schaltantrag
WFM	Workforcemanagement
CRM	Customer Relationship Management
ERP	Enterprise Resource Planning Software
MRS	Multi Resource Scheduling
SAN	Schaltantrag
SLT	Schutz und Leittechnik
GPRS	General Packet Radio Service
ZF	Zielfunktion
NB	Nebenbedingung
NNB	Nichtnegativitätsbedingung
OA	Optimierungsaufgabe
LOA	Lineare Optimierungsaufgabe
GLOA	Ganzzahlige lineare Optimierungsaufgabe
DOA	Diskrete Optimierungsaufgabe
GOA	Ganzzahlige Optimierungsaufgabe
TSP	Traveling Salesman Problem
VRP	Vehicle Routing Problem
DVRP	Distance constraint Vehicle Routing Problem
AZ	Auftragszuordnung
TT	Tagestouren

Notationen

$\|.\|$	Mächtigkeit einer Menge
X^*	optimale Lösung
R^+	positive reelle Zahlen
M_∞	hinreichend großer Wert
σ	Streuung
c_{ij}	Kosten zwischen i und j
$M = \{m_1, m_2, ..., m_{\|M\|}\}$	Menge der Aufträge
$\|M\|$	Anzahl der Aufträge
m_j	j-ter Auftrag
M_j	Menge der Aufträge, die dem Monteur n_j bei der Auftragszuordnung zugeordnet wurden
p_{m_j}	Summe der Plandauer der Vorgänge zum Auftrag m_j
$N = \{n_1, n_2, ..., n_{\|N\|}\}$	Menge der Monteure
$\|N\|$	Menge der Anzahl der Monteure
n_i	i.ter Monteur
$d(m_j, n_i)$	Luftlinienentfernung zwischen Ausführungsort des Auftrags m_i und Startadresse n_i
$d(v_i, v_j)$	Routenentfernung zwischen Ausführungsort des Vorganges v_i und v_j
$V = \{v_1, v_2..., v_{\|V\|}\}$	Menge der Hauptvorgänge
$NV = \{v_{\|V\|+1}, v_{\|V\|+2}, ..., v_{\|V\|+\|NV\|}\}$	Menge der Nebenvorgänge
$RG(i)$	Regret zum Mittel i
s_{ij}	Saving zwischen i und j
l_{ij}	zeitliche Lücke zwischen v_i und v_j
M^s	Menge der Splits
T_k	Tour/Tagestour k
$[s^j, e^j]$	Ausführungszeitraum des Vorganges v_j

Abbildungsverzeichnis

Literaturverzeichnis

[1] *envia NSG: Begriffe Netzbetrieb- TR 5-PUB 1ß.920/100*. betriebliches Dokument, 01.01.2008.

[2] *envia NSG: Pflichtenheft: Implementierung einer Optimierungsfunktion in ein WFM-System*. betriebliches Dokument, 2009.

[3] *PSI/enviaNSG: Lastenheft zur automatischen Auftragseinplanung und Vordisposition in RessMa*. betriebliches Dokument, Februar 2009.

[4] *mobileX AG: http://www.mobilexag.de/bigace/Was-ist-WFM.pdf* . Aufgerufen am 18.09.2009.

[5] *Siemens: http://www.pse.siemens.at/apps/sis/ge/gseinternet.nsf*. Aufgerufen am 18.09.2009

[6] *C. Mohan, K. Deep: Optimization Techniques*. New Age Science, 2009.

[7] *R. Fischer: Vorlesungsmitschrift: Diskrete Optimierung*. SS 2009.

[8] *P. Stingl: Operations Research Linearoptimierung*. Fachbuchverlag Leipzig, 2002.

[9] *F. Jarre, J. Stoer: Optimierung*. Springer-Verlag Berlin Heidelberg, 2005.

[10] *P. Tittmann: Graphentheorie*. Fachbuchverlag Leipzig, 2003.

[11] *D. L. Applegate, R. E. Bixby, V. Chvátal, W. J. Cook: The Traveling Salesman Problem: A Computational Study* . Princeton University Press, 2006.

[12] *W. Domschke, A. Drexl: Einführung in Operations Research*. Springer Verlag, 6.Auflage, 2005, Seite 142ff.

[13] *M. Kämpf: Probleme der Tourenbildung*. Chemnitzer Informatik-Berichte, TU Chemnitz, Mai 2006.

[14] *G. Ghiani, G. Laporte, R. Musmanno: Indroduction to Logistics Systems Planning ans Control*. Kapitel 7, Wiley, 2004.

[15] *M. Reuther: Tourenplanung mit Längenbeschränkung*. Diplomarbeit, Hochschule Mittweida, 2008.

[16] *J. Rieck: Tourenplanung mittelständischer Speditionsunternehmen*. Dissertation, TU Clauthal, 2008.

[17] *L. Fischer, M. Dod, C. Harth: Lösungsstrategien für ein Mehrdepot-Tourenplanungsproblem*. Hochschule Mittweida, Projektarbeit 2010.

[18] *U. Bankhofer, M. Wilhelm, G. Williner: Modelle und Methoden der Tourenplanung*. Arbeitsbericht, TU Illmenau, 2006.

[19] *G. Appa, L.Pitsoulis, H. P. Williams: Handbook on Modelling for diskrete Optimization*. Kapitel 5, Springer, 2006.

[20] *Alexander Mehlmann: http://www.eos.tuwien.ac.at/OR/Mehlmann/Andis/publ/or1/mopvorl4.pdf*. Vorlesungsskript Operations Research, TU Wien, Aufgerufen am 02.03.2010.

[21] *Schreier: http://www.mathe.tu-freiberg.de/ schreier/Transport/TPO11.pdf*. TU Freiberg, aufgerufen am 15.03.2010.

[22] *G. B. Dantzig, M. N. Thapa: Linear Programming 2: Theory and Extensions*. Springer-Verlag, 2003.

[23] *C. Heppner: Tabu Search*. Skript, TU Hamburg, 2005.

[24] *P. Becker: http://www2.inf.fh-rhein-sieg.de/ pbecke2m/or/tabu1.pdf*, Vorlesungsskript Operations Research, FH Rhein, aufgerufen am 03.03.2010.

[25] *L. Suhl, T. Mellouli: Optimierungssysteme*. Kapitel 5 & 8, Springer, 2009.

[26] *M. Schröder: Gebiete optimal aufteilen*. Dissertation, Karlsruhe, 2001.

[27] *www.fuzzy.de* aufgerufen am 10.10.2009.

[28] *www.clicksoftware.com* aufgerufen am 10.10.2009.

[29] *www.fls.info* aufgerufen am 10.10.2009

[30] *RWE: Disposition mit FLS-VISITOUR* Besprechungsprotokoll vom 11.05.2009.

[31] *www.sage.deOffice-Line-Cockpit* aufgerufen am 10.10.2009.

[32] *www.gts-systems.de/tourenplanung* aufgerufen am 10.10.2009.

[33] *www.sap.com/germany/index.epx* aufgerufen am 10.10.2009.

[34] *Marika Kästner: Sensitivitätsanalyse und Marktrecherche*, betriebliches Dokument, 2009.